

ĐẠI HỌC BÁCH KHOA HÀ NỘI



ĐỒ ÁN TỐT NGHIỆP CỦ NHÂN

**Thiết kế và triển khai hệ thống cảm biến PIR ứng dụng
trong Smarthome**

NGUYỄN VĂN MÙNG

Mung.nv212894@sis.hust.edu.vn

Ngành Kỹ thuật Điều khiển và Tự động hóa

Giảng viên hướng dẫn: PGS. TS. Nguyễn Quốc Cường

Khoa: Tự động hóa

Chữ ký của GVHD

HÀ NỘI, 1/2025

NHIỆM VỤ
ĐỒ ÁN TỐT NGHIỆP CỦ NHÂN

Họ và tên sinh viên: Nguyễn Văn Mừng

Khóa :K66 Trưởng :Điện-Điện tử Nghành: KTĐK & TĐH

1. *Tên đề tài*

Thiết kế và triển khai hệ thống cảm biến PIR ứng dụng trong Smarthome

2. *Nội dung đề tài*

Thiết kế và triển khai một hệ thống cảm biến PIR (Passive Infrared Sensor) ứng dụng trong Smarthome, có khả năng phát hiện và theo dõi chuyển động của con người. Hệ thống bao gồm mạch khuếch đại và lọc tín hiệu, kết nối WiFi, truyền dữ liệu đến nền tảng Thingsboard, đồng thời có kiến trúc mở để tích hợp các thuật toán tinyML nâng cao trong tương lai.

Các công việc bao gồm:

- Thiết kế mạch khuếch đại và lọc tín hiệu PIR, đảm bảo hệ thống hoạt động ổn định trong điều kiện môi trường thay đổi (nhiệt độ, độ ẩm).
- Sử dụng vi điều khiển ESP32 để thực hiện quá trình số hóa tín hiệu từ cảm biến PIR thông qua ADC (Analog-to-Digital Converter).
- Lập trình kết nối WiFi và truyền dữ liệu thời gian thực lên nền tảng IoT Thingsboard để lưu trữ và hiển thị.
- Thử nghiệm thuật toán phát hiện hoặc theo dõi chuyển động của một hoặc nhiều người dựa trên dữ liệu từ các cảm biến PIR.
- Đảm bảo hệ thống có kiến trúc mở để các nhóm có thể tích hợp thêm các mô-đun tinyML nhằm triển khai các thuật toán xử lý tín hiệu nâng cao sau này.

3. Thời gian giao đề tài :09/2024

4. Thời gian hoàn thành: 01/2025

Ngày 20 tháng 01 năm 2024
Cán bộ hướng dẫn

PGS.TS Nguyễn Quốc Cường

Lời cảm ơn

Em xin chân thành cảm ơn **PGS.TS Nguyễn Quốc Cường** đã tận tình hướng dẫn, chỉ bảo và đưa ra những nhận xét cụ thể, chi tiết trong suốt quá trình thực hiện đồ án. Sự hỗ trợ và định hướng của thầy đã giúp em hoàn thành đồ án này một cách tốt nhất.

Bên cạnh đó, em cũng xin gửi lời cảm ơn đến các thầy cô, anh chị và bạn bè trong Sensor Lab đã tạo điều kiện thuận lợi và hỗ trợ em trong suốt quá trình học tập và thực hiện đồ án.

Do kiến thức còn hạn chế, nên trong đồ án không tránh khỏi những thiếu sót và sai sót. Em rất mong nhận được những ý kiến đóng góp từ thầy cô và các bạn để giúp em hoàn thiện hơn trong các dự án sau này.

Em xin chân thành cảm ơn!

Hà Nội, ngày.... tháng.... năm

Sinh viên thực hiện

Nguyễn Văn Mừng

Tóm tắt nội dung đồ án

Đồ án "Thiết kế mảng cảm biến PIR - ứng dụng trong Smarthome" tập trung xây dựng hệ thống cảm biến PIR phát hiện và theo dõi chuyển động của con người, ứng dụng trong nhà thông minh. Hệ thống bao gồm mạch khuếch đại và lọc tín hiệu, sử dụng vi điều khiển ESP32 để số hóa tín hiệu qua ADC, lập trình kết nối WiFi và truyền dữ liệu thời gian thực. Ngoài ra, đồ án triển khai thuật toán phát hiện/chuyển động cơ bản trên thiết bị nhúng và thiết kế hệ thống với kiến trúc mở để tích hợp các thuật toán xử lý tín hiệu nâng cao như tinyML trong tương lai. Kết quả đạt được là một hệ thống cảm biến hoàn chỉnh, hoạt động ổn định trong các điều kiện môi trường khác nhau, đảm bảo tính chính xác, độ tin cậy và khả năng mở rộng. Đồ án mang tính thực tiễn cao, phù hợp với nhu cầu nhà thông minh hiện đại. Sinh viên đã phát triển các kỹ năng thiết kế phần cứng, lập trình nhúng, kết nối IoT và xây dựng hệ thống mở, tạo nền tảng cho việc nghiên cứu và ứng dụng các công nghệ tiên tiến.

Đồ án bao gồm 5 chương chính với nội dung như sau:

- Chương 1: Giới thiệu
- Chương 2: Cơ sở lý thuyết
- Chương 3: Thiết kế hệ thống
- Chương 4: Thực nghiệm đánh giá
- Chương 5: Kết luận và hướng phát triển

MỤC LỤC

CHƯƠNG 1. GIỚI THIỆU	1
1.1 Tổng quan về Smarthome.....	1
1.2 Cảm biến PIR và ứng dụng trong Smarthome.....	3
1.3 Mục tiêu của đồ án.....	4
1.4 Phạm vi thực hiện.....	4
1.5 Giải pháp đề xuất.....	5
1.6 Kết luận chương.....	6
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT.....	7
2.1 Nguyên lý hoạt động của cảm biến PIR	7
2.1.1 Khái niệm về hồng ngoại	7
2.1.2 Nguyên lý hoạt động của cảm biến PIR [1].....	8
2.1.3 Thấu kính Fresnel hội tụ và tập trung tia hồng ngoại.....	10
2.1.4 Các yếu tố ảnh hưởng tới độ nhạy của cảm biến PIR	11
2.2 Xử lý tín hiệu từ cảm biến PIR.....	12
2.2.1 Khuếch đại tín hiệu yếu từ cảm biến	12
2.2.2 Lọc tín hiệu để loại bỏ nhiễu và tín hiệu không mong muốn....	13
2.2.3 Phân tích tín hiệu ADC.....	14
2.3 Hệ thống IoT trong Smarthome.....	16
2.3.1 Các giao thức truyền thông phổ biến	16
2.3.2 Thingsboard [3]	18
2.4 Kiến trúc mở và khả năng tích hợp tinyML.....	21
2.4.1 Tổng quan về tinyML và ứng dụng trong phân tích tín hiệu cảm biến.	21
2.4.2 Lợi ích của kiến trúc mở trong các hệ thống TinyML [4]	22
CHƯƠNG 3. THIẾT KẾ HỆ THỐNG.....	24
3.1 Mô hình hệ thống	24
3.2 Lựa chọn linh kiện.....	26
3.2.1 Cảm biến D203S.....	26
3.2.2 IC LM358	26
3.2.3 MCU ESP32.....	27
3.2.4 Mô phỏng tín hiệu từ cảm biến D203S	29
3.2.5 Mạch nguyên lý	33

3.3	Thiết kế phần mềm	34
3.4	Kiến trúc mở tích hợp tinyML	36
3.4.1	Chuẩn bị dữ liệu từ cảm biến để huấn luyện mô hình.....	37
3.4.2	Đề xuất hướng mở rộng để tích hợp tinyML trong tương lai....	42
CHƯƠNG 4. THỰC NGHIỆM ĐÁNH GIÁ.....		44
4.1	Thử nghiệm phần cứng.....	44
4.2	Thử nghiệm phần mềm IOT.....	45
4.3	Thử nghiệm thuật toán với dữ liệu thu được cho việc triển khai tinyML trong tương lai.....	48
4.4	Đánh giá hệ thống tổng thể	55
4.5	Khó khăn.....	57
4.6	Phân tích chi phí và tiềm năng triển khai.....	58
CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....		60
5.1	Kết luận.....	60
5.2	Hướng phát triển trong tương lai.....	60
TÀI LIỆU THAM KHẢO.....		62
PHỤ LỤC.....		63

DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT

Tù viết tắt	Tên đầy đủ
IoT	Internet of Things (Internet vạn vật)
AI	Artificial Intelligence (Trí tuệ nhân tạo)
PIR	Passive Infrared (Cảm biến hồng ngoại thụ động)
tinyML	Tiny Machine Learning (Máy học nhỏ gọn)
ADC	Analog-to-Digital Converter (Bộ chuyển đổi tín hiệu tương tự sang số)
MQTT	Message Queuing Telemetry Transport
HTTP	Hypertext Transfer Protocol
AC	Alternating Current (Dòng điện xoay chiều)
DC	Direct Current (Dòng điện một chiều)
EMI	Electromagnetic Interference (Can nhiễu điện từ)
Op-amp	Operational Amplifier
LPF	Low Pass Filter (Bộ lọc thông thấp)
HPF	High Pass Filter (Bộ lọc thông cao)
SSL	Secure Sockets Layer (Lớp giao thức bảo mật)
TLS	Transport Layer Security (Bảo mật tầng vận chuyển)
CoAP	Constrained Application Protocol
SMS	Short Message Service
AWS	Amazon Web Services (Dịch vụ điện toán đám mây của Amazon)
MCU	Microcontroller Unit (Đơn vị vi điều khiển)
IMU	Inertial Measurement Unit (Đơn vị đo lường quán tính)
JSON	JavaScript Object Notation (Định dạng đối tượng JavaScript)
NTP	Network Time Protocol (Giao thức thời gian mạng)
DIP	Dual In-line Package

SOIC/SOP	Small Outline Integrated Circuit / Small Outline Package
I ² C	Inter-Integrated Circuit
UART	Universal Asynchronous Receiver-Transmitter
SPI	Serial Peripheral Interface
I2S	Inter-IC Sound
MAC	Media Access Control
DMA	Direct Memory Access
GPIO	General Purpose Input/Output
SSID	Service Set Identifier

DANH MỤC HÌNH VẼ

Hình 1.1 Các công nghệ trong Smarthome	1
Hình 2.1 Phổ điện từ.....	7
Hình 2.2.Cảm biến D203s	9
Hình 2.3.Cấu tạo bên trong của D203s.....	9
Hình 2.4.Nguyên lý hoạt động.....	10
Hình 2.5 Thấu kính Fresnel	11
Hình 2.6 Giao thức MQTT	16
Hình 2.7 Giao thức HTTP	17
Hình 2.8 Giao diện nền tảng Thingsboard	19
Hình 2.9 Kiến trúc IoT tích hợp TinyML và Điện toán biên (Edge Computing)	21
Hình 3.1 Mô hình hệ thống.....	24
Hình 3.2 Cảm biến D203S.....	26
Hình 3.3 Pin connections LM358.....	27
Hình 3.4.Modul ESP32	28
Hình 3.5.Mô phỏng trên LTSpice.....	30
Hình 3.6.Tín hiệu giả lập.....	30
Hình 3.7.Tầng khuỷch đại thứ nhất.....	31
Hình 3.8.Tín hiệu sau tầng khuỷch đại thứ nhất	31
Hình 3.9.Tầng khuỷch đại thứ hai.....	32
Hình 3.10.Tín hiệu ở tầng khuỷch đại thứ hai	32
Hình 3.11.Mạch nguyên lý	33
Hình 3.12.Mạch PCB	33
Hình 3.13.Mạch sau khi đã hoàn thiện.....	33
Hình 3.14.PIR Array.....	34
Hình 3.15.Cấu trúc gói tin gửi lên Thingsboard	35
Hình 3.16 Dual-Core trên ESP32	35
Hình 3.17 Giao diện Thingsboard trực quan hóa dữ liệu.....	36
Hình 3.18 Sơ đồ kiến trúc mờ tích hợp tinyML	37
Hình 3.19.Chiều cao từ nút cảm biến tới mặt sàn.....	37
Hình 3.20.Vùng giám sát của cảm biến.....	38
Hình 3.21.Bố trí cảm biến tương ứng các hướng.....	38
Hình 3.22.Quỹ đạo thu data.....	39
Hình 3.23.Quỹ đạo testcase	39
Hình 3.24.Giao diện web quay,thu và trực quan lúc thu.....	40
Hình 3.25 Giao diện gán nhãn.....	41
Hình 3.26.Cấu trúc file.csv.....	41

Hình 3.27.Dạng tín hiệu của trường hợp đi ở cột thứ hai	42
Hình 4.1 Tín hiệu của mảng cảm biến khi chưa phát hiện chuyển động	44
Hình 4.2 Tín hiệu của mảng cảm biến phát hiện có chuyển động trong khu vực giám sát.....	44
Hình 4.3 Mô hình hệ thống.....	45
Hình 4.4 Đồ thị trực quan trên Thingsboard khi chưa phát hiện chuyển động ...	46
Hình 4.5 Đồ thị trực quan trên Thingsboard khi phát hiện chuyển động trong vùng giám sát.....	46
Hình 4.6 Xét ngưỡng phát hiện người cho cảm biến.....	47
Hình 4.7 Thông báo phát hiện người trên Thingsboard.....	47
Hình 4.8.Các gói tin khi kết nối tốt.....	48
Hình 4.9.Mất gói tin khi kết nối kém.....	48
Hình 4.10 Training Loss.....	49
Hình 4.11 Kết quả kịch bản 1	50
Hình 4.12 Kết quả kịch bản 2	51
Hình 4.13 Kết quả kịch bản 3	51
Hình 4.14 Kết quả kịch bản 4	52
Hình 4.15 Training Loss.....	52
Hình 4.16 Phổ dự đoán vị trí.....	53
Hình 4.17 Kết quả kịch bản 1	54
Hình 4.18 Kết quả kịch bản 2	54
Hình 4.19 Kết quả kịch bản 3	55
Hình 4.20 Kết quả kịch bản 4	55
Hình 4.21 Dòng điện của mỗi cảm biến	56
Hình 4.22 Tín hiệu bị nhiễu do sóng điện từ.....	58
Hình 4.23 Tín hiệu output của cảm biến đo được khi bị nhiễu.....	58

DANH MỤC BẢNG BIỂU

Bảng 1 So sánh MQTT và HTTP	18
Bảng 2 Kết quả dự đoán theo từng nhãn	50

Kế hoạch thực hiện

Tuần	Giai đoạn	Chi tiết	Trạng thái
5,6	Nghiên cứu và phân tích yêu cầu công nghệ	<p>Tìm hiểu công nghệ cảm biến PIR.</p> <p>-Phân tích các yêu cầu phần cứng và phần mềm.</p>	Đã hoàn thành
7,8	Thiết kế mạch lọc và khuếch đại	<p>-Thiết kế mạch khuếch đại tín hiệu từ cảm biến PIR.</p> <p>-Mô phỏng lại tín hiệu và so sánh nó với kết quả rút ra từ đồ án 1</p>	Đã hoàn thành
9,10	Số hóa tín hiệu và kết nối ESP32	<p>Đọc giá trị ADC từ cảm biến</p> <p>-Kết nối Wifi</p> <p>-Làm vỏ cho PIR</p>	Đã hoàn thành
11,12	Triển khai Thingsboard và xử lý dữ liệu	<p>-Truyền dữ liệu lên</p> <p>-Thingsboard, trực quan hóa dữ liệu và lưu vào cơ sở dữ liệu, phục vụ cho AI</p> <p>-Thu data</p>	Đã hoàn thành

12,14,15,16,17	<p>-Tích hợp và tối ưu thuật toán tinyML</p>	<p>-Triển khai thuật toán phát hiện hoặc theo dõi chuyển động của một hoặc nhiều người dựa trên dữ liệu từ các cảm biến PIR.</p> <p>-Hệ thống có kiến trúc mở để các nhóm có thể tích hợp thêm các mô-đun tinyML nhằm triển khai các thuật toán xử lý tín hiệu nâng cao sau này.</p>	Đã hoàn thành
----------------	--	--	---------------

CHƯƠNG 1. GIỚI THIỆU

1.1 Tổng quan về Smarthome

Trong bối cảnh công nghệ thông minh ngày càng phổ biến và dễ tiếp cận, "Smarthome" đã trở thành một trong những ứng dụng nổi bật, nhanh chóng lan tỏa trên toàn cầu, định hình tương lai của cuộc sống hiện đại. Smarthome là hệ sinh thái các thiết bị và hệ thống được kết nối thông qua IoT (Internet of Things), không chỉ tự động hóa các thao tác cơ bản như bật tắt đèn, điều chỉnh nhiệt độ, mà còn có khả năng nhận biết, học hỏi và dự đoán hành vi của người dùng. Với sự hỗ trợ từ trí tuệ nhân tạo (AI) và phân tích dữ liệu lớn (Big Data), Smarthome mang lại trải nghiệm sống thông minh, tiện nghi, an toàn và tiết kiệm năng lượng hơn bao giờ hết. Đặc biệt, công nghệ cảm biến theo dõi chuyển động đóng vai trò cốt lõi trong việc tự động hóa điều khiển thiết bị, đảm bảo an ninh và tối ưu hóa năng lượng. Các cảm biến này không chỉ giúp bật tắt đèn tự động theo sự hiện diện của con người, mà còn kích hoạt hệ thống báo động khi phát hiện các hoạt động bất thường, góp phần nâng cao mức độ an toàn cho ngôi nhà. Ngoài ra, việc ứng dụng cảm biến chuyển động vào quản lý năng lượng giúp giảm lãng phí, tối ưu chi phí vận hành các thiết bị điện. Cùng với khả năng điều khiển từ xa qua ứng dụng di động hoặc giọng nói, Smarthome không chỉ là giải pháp tiện nghi mà còn là bước tiến lớn hướng tới lối sống bền vững, thông minh và hiệu quả. Với sự phát triển mạnh mẽ của AI, IoT và công nghệ 5G, Smarthome hứa hẹn sẽ tiếp tục cách mạng hóa không gian sống, biến ngôi nhà thành một hệ sinh thái thông minh, đáp ứng tối đa mọi nhu cầu của con người.



Hình 1.1 Các công nghệ trong Smarthome

Các công nghệ theo dõi chuyển động trong Smarthome

Hiện nay, trên thị trường có nhiều công nghệ theo dõi chuyển động khác nhau được ứng dụng trong Smarthome, bao gồm:

-Camera giám sát thông minh:

Đây là giải pháp phổ biến nhất, sử dụng camera có tích hợp AI để nhận diện chuyển động, phân biệt con người với các vật thể khác, và thậm chí nhận diện khuôn mặt. Các camera này có thể gửi cảnh báo tức thời hoặc kích hoạt các thiết bị khác như đèn hoặc còi báo động khi phát hiện chuyển động bất thường. Tuy nhiên, hệ thống này có một số nhược điểm:

- Yêu cầu băng thông lớn để truyền tải hình ảnh/video, đặc biệt trong các ứng dụng thời gian thực.
- Chi phí cao và tốn nhiều năng lượng.
- Lo ngại về quyền riêng tư, vì dữ liệu hình ảnh có thể bị xâm phạm hoặc lạm dụng.

Radar Doppler

Công nghệ radar sử dụng sóng vô tuyến để phát hiện chuyển động thông qua hiệu ứng Doppler. Ưu điểm của radar là khả năng làm việc trong mọi điều kiện ánh sáng, kể cả ban đêm hoặc trong môi trường khói bụi. Tuy nhiên, radar có chi phí triển khai cao và thường chỉ phù hợp với các ứng dụng đặc thù, như giám sát an ninh cấp cao hoặc công nghiệp.

Cảm biến siêu âm (Ultrasonic):

Cảm biến này hoạt động bằng cách phát ra sóng siêu âm và phân tích sóng phản xạ để phát hiện chuyển động. Giải pháp này tương đối rẻ và có độ chính xác cao trong phạm vi gần. Tuy nhiên, cảm biến siêu âm dễ bị nhiễu bởi các vật cản trong môi trường và thường không thích hợp để theo dõi chuyển động phức tạp hoặc từ nhiều người cùng lúc.

Cảm biến PIR (Passive Infrared Sensor):

Cảm biến PIR hoạt động bằng cách phát hiện tia hồng ngoại phát ra từ cơ thể người. Với ưu điểm tiết kiệm năng lượng, chi phí thấp, và không phát ra tín hiệu chủ động, cảm biến PIR phù hợp với các ứng dụng Smarthome. Tuy nhiên, cảm biến PIR truyền thông có hạn chế về khả năng phân biệt các đối tượng hoặc nhận diện hướng chuyển động, đòi hỏi phải có sự hỗ trợ từ các giải pháp xử lý tín hiệu để nâng cao hiệu quả hoạt động.

Ưu thế của hệ thống cảm biến PIR trong đề tài

So với các công nghệ khác, hệ thống cảm biến PIR được thiết kế trong đề tài "Thiết kế mảng cảm biến PIR - ứng dụng trong Smarthome" mang lại nhiều ưu điểm vượt trội trong môi trường Smarthome:

- **Tiết kiệm năng lượng:** Không giống như camera hoặc radar, cảm biến PIR không cần tiêu thụ nhiều năng lượng vì không phát tín hiệu chủ động. Điều này đặc biệt phù hợp với các thiết bị chạy pin hoặc các hệ thống yêu cầu tính ổn định cao.

- **Chi phí thấp:** So với camera hoặc radar, cảm biến PIR rẻ hơn nhiều, giúp giảm thiểu chi phí lắp đặt và bảo trì cho người dùng.
- **Tính bảo mật cao:** Vì chỉ sử dụng tín hiệu hồng ngoại, hệ thống này không lưu trữ hoặc truyền tải hình ảnh/video, qua đó đảm bảo quyền riêng tư cho người dùng.

1.2 Cảm biến PIR và ứng dụng trong Smarthome

Cảm biến PIR (Passive Infrared Sensor) là một trong những công nghệ quan trọng và phổ biến nhất được sử dụng trong các hệ thống Smarthome, đặc biệt trong việc nhận diện và theo dõi chuyển động. Cảm biến PIR hoạt động dựa trên nguyên lý phát hiện sự thay đổi bức xạ hồng ngoại trong môi trường, giúp phát hiện sự di chuyển của con người hoặc động vật trong phạm vi cảm biến. Đây là một giải pháp hiệu quả về chi phí và năng lượng, góp phần nâng cao khả năng tự động hóa và bảo mật trong ngôi nhà thông minh.

Một trong những ứng dụng phổ biến nhất là tự động hóa điều khiển thiết bị, nơi cảm biến PIR được tích hợp với hệ thống chiếu sáng để tự động bật/tắt đèn khi phát hiện chuyển động, giúp tiết kiệm năng lượng và tăng cường sự tiện nghi. Ví dụ, đèn hành lang hoặc nhà vệ sinh sẽ tự động sáng khi có người bước vào và tắt khi không còn chuyển động. Trong lĩnh vực an ninh, cảm biến PIR được sử dụng để phát hiện xâm nhập trái phép, khi phát hiện chuyển động bất thường, nó sẽ gửi cảnh báo đến hệ thống hoặc kích hoạt camera giám sát để ghi lại hình ảnh, góp phần bảo vệ an toàn cho ngôi nhà. Bên cạnh đó, một số hệ thống Smarthome sử dụng cảm biến PIR để phát hiện sự hiện diện của con người trong phòng, từ đó điều chỉnh hoạt động của điều hòa nhiệt độ hoặc quạt thông minh nhằm tối ưu hóa năng lượng. Ngoài ra, cảm biến PIR có thể tích hợp với các trợ lý ảo như Alexa hoặc Google Assistant, cho phép người dùng điều khiển thiết bị bằng giọng nói hoặc thông qua ứng dụng di động, mang lại sự linh hoạt và tiện lợi trong cuộc sống hàng ngày.

Mặc dù mang lại nhiều lợi ích, cảm biến PIR vẫn tồn tại một số hạn chế khi áp dụng vào các hệ thống Smarthome hiện đại. Một trong những thách thức lớn nhất là cảm biến PIR nhạy cảm với nhiệt độ môi trường, khi nhiệt độ xung quanh cao gần bằng nhiệt độ cơ thể người, khả năng phát hiện chuyển động sẽ giảm do sự chênh lệch bức xạ hồng ngoại thấp. Bên cạnh đó, cảm biến này đôi khi gặp vấn đề với các chuyển động giả, chẳng hạn như cành cây lay động, vật nuôi di chuyển, hoặc luồng gió ám từ điều hòa, dẫn đến các cảnh báo sai trong hệ thống an ninh. Ngoài ra, cảm biến PIR có phạm vi và góc quét giới hạn, khiến các góc chết vẫn tồn tại trong một số trường hợp, buộc người dùng phải lắp đặt nhiều cảm biến để giám sát toàn bộ khu vực, làm tăng chi phí. Hơn nữa, cảm biến PIR chỉ phát hiện được chuyển động mà không thể nhận diện các vật thể đứng yên, kể cả con người, điều này làm giảm hiệu quả trong các hệ thống yêu cầu nhận diện liên tục. Cuối cùng, khả năng tương thích của một số cảm biến PIR truyền thống với các hệ thống thông minh hiện đại hoặc các giao thức kết nối như Zigbee, Z-Wave, hay Wi-Fi vẫn còn hạn chế, gây khó khăn trong việc đồng bộ hóa với các thiết bị khác.

Chính vì những hạn chế kể trên, tôi đề xuất thực hiện một đồ án thiết kế nhằm khắc phục và nâng cao hiệu quả ứng dụng của cảm biến PIR trong các hệ thống Smarthome. Đồ án bao gồm việc xây dựng **một hệ thống mảng cảm biến PIR hoàn chỉnh**, tích hợp cả phần cứng và phần mềm, với khả năng đo lường tín hiệu chuyển động và truyền dữ liệu đến nền tảng Thingsboard để quản lý và theo dõi thông minh. Đồng thời, hệ thống sẽ được đánh giá về độ chính xác và độ tin cậy trong các điều kiện môi trường khác nhau nhằm đảm bảo hiệu suất tối ưu. Cuối cùng, kiến trúc hệ thống sẽ được thiết kế mở để hỗ trợ tích hợp các thuật toán **tinyML**, giúp tăng cường khả năng nhận diện và phân tích chuyển động, đồng thời mở ra tiềm năng ứng dụng các giải pháp trí tuệ nhân tạo trong tương lai.

1.3 Mục tiêu của đồ án

1.2.1 Mục tiêu đề tài:

Mục tiêu của đề tài là thiết kế và triển khai một hệ thống cảm biến PIR (Passive Infrared Sensor) ứng dụng trong Smarthome, có khả năng phát hiện và theo dõi chuyển động của con người. Hệ thống bao gồm mạch khuếch đại và lọc tín hiệu, kết nối WiFi, truyền dữ liệu đến nền tảng Thingsboard, đồng thời có kiến trúc mở để tích hợp các thuật toán tinyML nâng cao trong tương lai.

1.2.2 Nhiệm vụ của sinh viên:

- Thiết kế mạch khuếch đại và lọc tín hiệu PIR, đảm bảo hệ thống hoạt động ổn định trong điều kiện môi trường thay đổi (nhiệt độ, độ ẩm).
- Sử dụng vi điều khiển ESP32 để thực hiện quá trình số hóa tín hiệu từ cảm biến PIR thông qua ADC (Analog-to-Digital Converter).
- Lập trình kết nối WiFi và truyền dữ liệu thời gian thực lên nền tảng IoT Thingsboard để lưu trữ và hiển thị.
- Triển khai thuật toán phát hiện hoặc theo dõi chuyển động của một hoặc nhiều người dựa trên dữ liệu từ các cảm biến PIR.
- Đảm bảo hệ thống có kiến trúc mở để các nhóm có thể tích hợp thêm các mô-đun tinyML nhằm triển khai các thuật toán xử lý tín hiệu nâng cao sau này.

1.2.3 Kết quả mong đợi:

- Một hệ thống mảng cảm biến PIR hoàn chỉnh, bao gồm phần cứng và phần mềm, có khả năng đo lường tín hiệu chuyển động và truyền dữ liệu đến nền tảng Thingsboard.
- Đánh giá độ chính xác và độ tin cậy của hệ thống trong điều kiện môi trường khác nhau.
- Thuật toán phát hiện hoặc theo dõi chuyển động cơ bản được triển khai trên thiết bị nhúng.
- Kiến trúc hệ thống mở cho phép mở rộng tích hợp các thuật toán tinyML.

1.4 Phạm vi thực hiện

a. Phần cứng và thiết kế mạch

- Thiết kế và phát triển hệ thống phần cứng mảng cảm biến PIR, bao gồm việc lựa chọn loại cảm biến, xây dựng các module khuếch đại và lọc tín hiệu để đảm bảo tín hiệu chính xác và hạn chế ảnh hưởng từ các yếu tố môi trường như nhiệt độ và độ ẩm.
- Thiết kế hệ thống kết nối và xử lý dữ liệu sử dụng ESP32 để nhận tín hiệu từ cảm biến, số hóa thông qua ADC và truyền dữ liệu qua Wi-Fi.

b. Phần mềm và nền tảng IoT

- Lập trình trên vi điều khiển ESP32 để quản lý mảng cảm biến, bao gồm đọc tín hiệu từ các cảm biến PIR, số hóa tín hiệu và phát hiện chuyển động.
- Tích hợp nền tảng IoT Thingsboard để hiển thị và lưu trữ dữ liệu thời gian thực từ hệ thống. Điều này giúp người dùng dễ dàng giám sát chuyển động trong ngôi nhà, đồng thời mở ra khả năng lưu trữ và phân tích dữ liệu.

c. Ứng dụng và thử nghiệm trong môi trường Smarthome

- Áp dụng hệ thống cảm biến PIR trong các tình huống Smarthome, như điều khiển thiết bị chiếu sáng tự động, kích hoạt camera an ninh hoặc gửi cảnh báo khi phát hiện chuyển động.
- Thực hiện các thử nghiệm về độ nhạy, độ chính xác của cảm biến và khả năng hoạt động ổn định của hệ thống trong các điều kiện môi trường khác nhau, như thay đổi nhiệt độ, độ sáng, và sự hiện diện của các vật thể khác.

d. Nâng cấp và mở rộng

- Tạo kiến trúc mở để hệ thống có thể mở rộng, cho phép tích hợp các thuật toán học máy (TinyML) nhằm nâng cao khả năng phát hiện các mẫu chuyển động phức tạp, dự đoán các hành vi và phân biệt các loại chuyển động khác nhau.
- Đưa ra các phương pháp để tích hợp thêm các cảm biến khác trong tương lai hoặc mở rộng quy mô ứng dụng cho nhiều phòng hoặc toàn bộ ngôi nhà, giúp nâng cao tính linh hoạt của hệ thống.

1.5 Giải pháp đề xuất

Để xây dựng một hệ thống cảm biến PIR hiệu quả trong ứng dụng Smarthome, giải pháp đầu tiên là thiết kế một mạch khuếch đại và lọc tín hiệu tối ưu. Mạch này sẽ giúp tăng cường tín hiệu từ cảm biến PIR và giảm thiểu nhiễu từ môi trường xung quanh, chẳng hạn như biến đổi nhiệt độ và độ ẩm. Bằng cách sử dụng các thành phần linh kiện chính xác, hệ thống có thể đảm bảo hoạt động ổn định trong nhiều điều kiện khác nhau.

Việc sử dụng vi điều khiển ESP32 đóng vai trò quan trọng trong giải pháp này. ESP32 không chỉ cung cấp khả năng số hóa tín hiệu từ cảm biến PIR thông qua bộ chuyển đổi ADC (Analog-to-Digital Converter) mà còn tích hợp khả năng kết nối WiFi, cho phép truyền dữ liệu thời gian thực đến các nền tảng IOT platform..

Ngoài chức năng cơ bản, hệ thống được thiết kế với khả năng mở rộng, cho phép tích hợp các thuật toán xử lý tín hiệu nâng cao dựa trên công nghệ tinyML. Các

thuật toán này có thể triển khai trực tiếp trên thiết bị để tăng cường khả năng nhận diện và theo dõi chuyển động phức tạp, chẳng hạn như phân biệt chuyển động của con người và vật nuôi. Điều này giúp hệ thống trở nên thông minh hơn và phù hợp hơn với các yêu cầu thực tế.

Để đảm bảo tính khả thi và hiệu quả, hệ thống sẽ được thử nghiệm trong các môi trường khác nhau, từ nhà ở thông thường đến các không gian có điều kiện môi trường khắc nghiệt hơn. Kết quả thử nghiệm sẽ được so sánh để đánh giá độ nhạy, độ tin cậy và hiệu suất của hệ thống, từ đó tiếp tục tối ưu hóa phần cứng và phần mềm.

Về mặt phần mềm, các giao thức truyền thông hiện đại như MQTT/HTTP sẽ được tích hợp để cải thiện tốc độ và độ ổn định của việc truyền dữ liệu. Đồng thời, giao diện trực quan trên các nền tảng IoT sẽ được thiết kế để người dùng dễ dàng theo dõi và điều khiển hệ thống cảm biến, mang lại trải nghiệm sử dụng tiện lợi và hiệu quả.

Giải pháp này không chỉ tập trung vào việc đáp ứng các yêu cầu cơ bản mà còn đặt nền móng cho việc phát triển các hệ thống nhà thông minh thế hệ mới. Với kiến trúc mở, hệ thống cảm biến có thể dễ dàng tích hợp với các công nghệ AIoT trong tương lai, như tự động học hành vi người dùng hay dự đoán các kịch bản trong nhà. Điều này không chỉ tăng cường sự tiện nghi mà còn góp phần tiết kiệm năng lượng và bảo vệ môi trường.

1.6 Kết luận chương

Đề tài "Thiết kế mảng cảm biến PIR - ứng dụng trong Smarthome" đã hoàn thành mục tiêu xây dựng một hệ thống giám sát chuyển động với tính năng kết nối IoT, mang lại nhiều lợi ích cho người dùng trong môi trường Smarthome. Kết quả đạt được cho thấy hệ thống có thể phát hiện chuyển động với độ chính xác cao, cung cấp dữ liệu theo thời gian thực và kết nối ổn định với nền tảng Thingsboard. Giải pháp này không chỉ giúp tối ưu hóa việc sử dụng năng lượng và tăng cường an ninh mà còn cho phép người dùng điều khiển và giám sát thiết bị từ xa, nâng cao sự tiện lợi và thoải mái trong sinh hoạt hàng ngày.

Hơn nữa, thiết kế của hệ thống đã được tối ưu để dễ dàng mở rộng, cho phép tích hợp các thuật toán TinyML trong tương lai nhằm phát triển các chức năng phân tích nâng cao, như phân biệt chuyển động của con người và vật nuôi. Ngoài ra, với khả năng bổ sung thêm các cảm biến khác, hệ thống có tiềm năng đáp ứng nhiều nhu cầu khác nhau trong các ứng dụng Smarthome và quản lý năng lượng thông minh.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

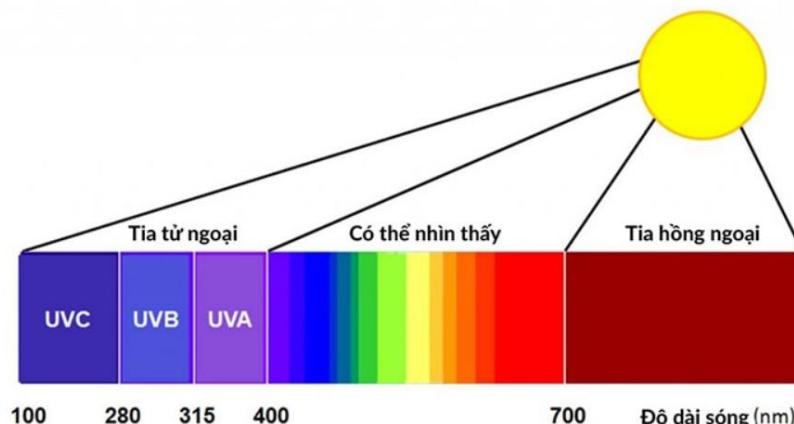
2.1 Nguyên lý hoạt động của cảm biến PIR

2.1.1 Khái niệm về hồng ngoại

Khi nhiệt độ của bất kỳ vật thể rắn nào cao hơn nhiệt độ tuyệt đối (-273°C hoặc 0 K), vật thể đó sẽ phát ra bức xạ điện từ, tạo nên một quang phổ bức xạ nhiệt. Đây là một hiện tượng tự nhiên đã được nghiên cứu và chứng minh bởi nhiều nhà khoa học. Wilhelm Wien đã thiết lập mối quan hệ quan trọng giữa nhiệt độ tuyệt đối của vật thể (T) và bước sóng phát xạ mạnh nhất (λ_{\max}) trong quang phổ thông qua công thức:

$$\lambda_{\max} = \frac{2898}{T}$$

Trong công thức này, T được đo bằng đơn vị Kelvin (K) và λ_{\max} được tính bằng micromet (μm). Ví dụ minh họa rõ nét cho mối quan hệ này là nhiệt độ bề mặt cơ thể người khoảng 37°C (tương đương 310K), phát ra bước sóng tối đa khoảng 9,4 μm . Tương tự, một con mèo với nhiệt độ cơ thể trung bình là 38°C (311K) sẽ phát ra bước sóng tối đa 9,3 μm . Điều này chứng minh rằng bất kỳ vật thể nào có nhiệt độ cao hơn 0 K đều phát ra bức xạ dưới dạng sóng điện từ, mà phổ phát xạ phụ thuộc trực tiếp vào nhiệt độ của vật thể.



Hình 2.1 Quang phổ

Phổ phát xạ của một vật thể không phải là cố định mà thay đổi theo nhiệt độ. Các vật thể có nhiệt độ càng cao sẽ phát ra bức xạ ở bước sóng càng ngắn và ngược lại. Chẳng hạn, ở nhiệt độ rất cao như 2000K, phần lớn năng lượng phát xạ sẽ nằm trong phổ ánh sáng nhìn thấy (ví dụ, ánh sáng đỏ hoặc trắng). Tuy nhiên, khi nhiệt độ giảm xuống, chẳng hạn ở mức 500K, phần lớn năng lượng phát xạ sẽ nằm trong phổ hồng ngoại – một loại bức xạ mà mắt người không thể nhìn thấy.

Chính vì đặc tính này, việc phát hiện bức xạ hồng ngoại từ các vật thể ở nhiệt độ thấp đòi hỏi sự hỗ trợ của các cảm biến chuyên dụng. Các cảm biến hồng ngoại, bao gồm cảm biến PIR, được thiết kế để phát hiện bức xạ hồng ngoại phát ra từ vật thể trong môi trường xung quanh. Cảm biến này hoạt động dựa trên nguyên lý đo lường và so sánh sự thay đổi cường độ bức xạ hồng ngoại trong không gian và thời gian, giúp nhận biết sự hiện diện hoặc chuyển động của các vật thể.

Tuy nhiên, phổ bức xạ hồng ngoại của các vật thể thông thường thường rất rộng và khó phân biệt nếu chỉ dựa trên phân tích phổ. Ví dụ, cơ thể con người và các vật nuôi như chó, mèo có nhiệt độ tương đương, dẫn đến phổ hồng ngoại phát ra của chúng cũng tương tự nhau. Do đó, các hệ thống cảm biến hồng ngoại cần được tối ưu hóa không chỉ để phát hiện bức xạ mà còn phải tích hợp các thuật toán để phân biệt tín hiệu hiệu quả hơn.

Cảm biến PIR là một trong những loại cảm biến hồng ngoại phổ biến nhất được sử dụng trong các hệ thống thông minh, đặc biệt là Smarthome, bởi khả năng phát hiện chuyển động dựa trên sự thay đổi trong bức xạ hồng ngoại. Chúng không chỉ nhạy bén mà còn có chi phí thấp và dễ dàng tích hợp vào các ứng dụng hiện đại. Sự phát triển của công nghệ cảm biến PIR đã mở ra những bước tiến lớn trong việc tự động hóa nhà ở, nâng cao an ninh, và tối ưu hóa năng lượng.

2.1.2 Nguyên lý hoạt động của cảm biến PIR [1]

Cảm biến PIR (Pyroelectric Infrared Sensor) hoạt động dựa trên nguyên lý cảm ứng pyroelectric, một hiện tượng vật lý trong đó các vật liệu pyroelectric như polyme hoặc gốm (keramik) tạo ra điện thế khi bị làm nóng hoặc làm lạnh. Đây là loại cảm biến đặc biệt có khả năng phát hiện tia hồng ngoại (nhiệt) phát ra từ cơ thể con người hoặc các vật thể sống, thường được sử dụng trong các hệ thống nhận diện chuyển động và an ninh.

Nguyên lý hoạt động của cảm biến PIR bắt đầu với việc sử dụng một ống kính Fresnel, hoặc các loại ống kính tương tự, để tập trung tia hồng ngoại từ môi trường xung quanh lên bề mặt của chip cảm biến bên trong. Ống kính Fresnel không chỉ giúp tăng độ nhạy mà còn mở rộng phạm vi và góc phát hiện của cảm biến, đảm bảo thu thập đầy đủ bức xạ hồng ngoại từ các nguồn nhiệt khác nhau trong vùng quan sát.

Khi môi trường không có sự vật chuyển động, nhiệt độ trong vùng cảm biến sẽ ổn định, dẫn đến tín hiệu không thay đổi. Tuy nhiên, khi có chuyển động, chẳng hạn như một người đi qua, nhiệt độ trong khu vực quan sát thay đổi. Các thay đổi này xảy ra vì cơ thể người phát ra bức xạ hồng ngoại mạnh hơn hoặc khác biệt so với môi trường. Những biến đổi nhiệt độ này được chuyển đổi thành tín hiệu điện nhờ vào vật liệu pyroelectric, là thành phần chính của chip cảm biến.

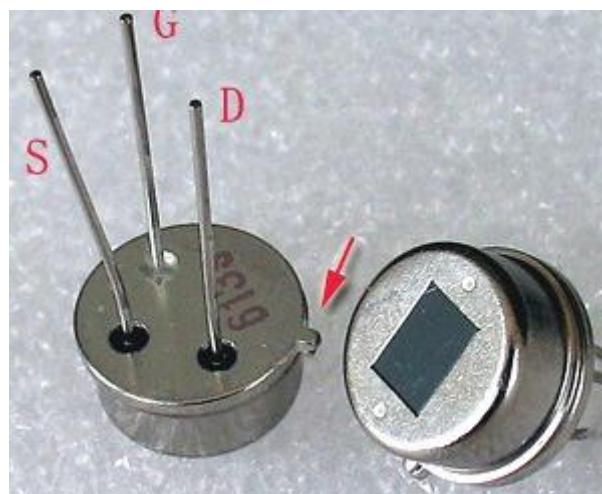
Quá trình chuyển đổi này diễn ra khi nhiệt độ thay đổi làm cấu trúc tinh thể của vật liệu pyroelectric bị phân cực, tạo ra một sự chênh lệch điện thế. Tín hiệu điện được tạo ra phản ánh mức độ thay đổi nhiệt trong vùng cảm biến. Tuy nhiên, tín hiệu này thường yếu và chứa nhiều nhiễu, nên cần được đưa vào một mạch xử lý tín hiệu.

Trong mạch xử lý, tín hiệu được lọc để loại bỏ các yếu tố gây nhiễu như sự thay đổi nhiệt độ môi trường chậm hoặc các nguồn nhiệt không mong muốn. Bộ lọc thông dải (band-pass filter) thường được sử dụng để chỉ cho phép các thay đổi nhiệt xảy ra ở tần số phù hợp với chuyển động của con người đi qua. Đồng thời, tín hiệu cũng được khuếch đại để đảm bảo đủ mạnh và dễ dàng xử lý bởi các thiết bị điều khiển tiếp theo.

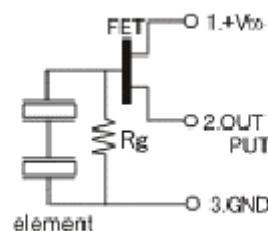
Đặc điểm nổi bật của cảm biến PIR là tính thụ động. Điều này có nghĩa là nó không phát ra bất kỳ dạng sóng năng lượng nào mà chỉ dựa vào việc thu nhận tia hồng ngoại từ môi trường, giúp tiết kiệm năng lượng và tăng độ bảo mật trong các ứng dụng. Cảm biến hoạt động liên tục để giám sát các thay đổi nhiệt, nhưng không can thiệp hay gây ảnh hưởng đến môi trường xung quanh.

Một khía cạnh quan trọng nữa là cách cảm biến được thiết kế với các vùng phát hiện (detection zones) thông qua ống kính Fresnel. Các vùng này được tổ chức theo dạng tổ ong, giúp cảm biến có khả năng phát hiện chuyển động từ nhiều góc độ khác nhau. Khi một vật thể có nhiệt độ cao hơn môi trường đi qua các vùng này, tín hiệu nhiệt dao động, phản ánh sự chuyển động và tốc độ của đối tượng.

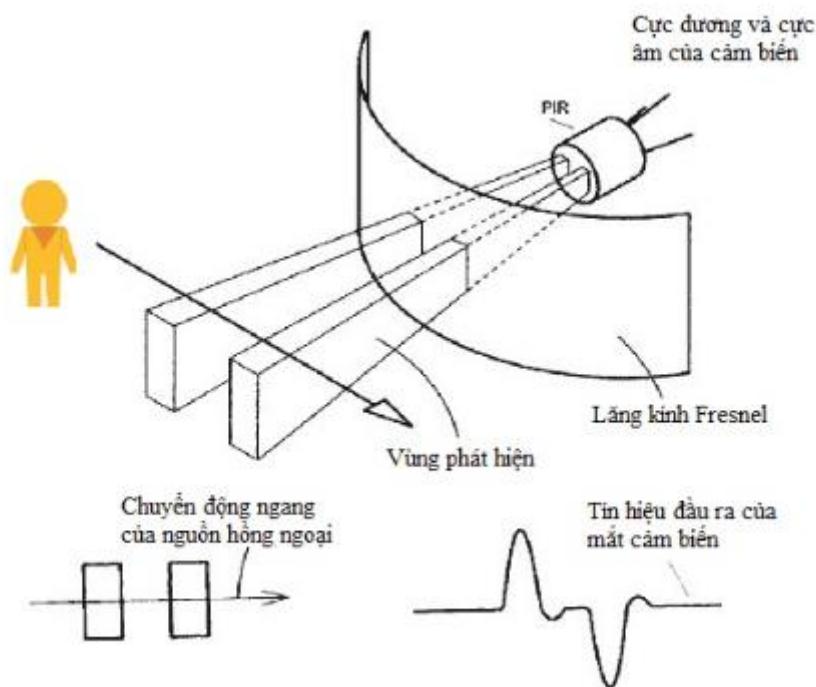
Nhờ thiết kế tối ưu và nguyên lý hoạt động như vậy, cảm biến PIR có thể phát hiện chuyển động chính xác trong nhiều điều kiện môi trường khác nhau. Tuy nhiên, để hoạt động hiệu quả, cảm biến cần được đặt ở vị trí phù hợp, tránh ánh sáng mặt trời trực tiếp hoặc các nguồn nhiệt không mong muốn, đồng thời phải được tích hợp với các mạch điện tử và phần mềm xử lý phù hợp để giảm thiểu sai sót và nhiễu tín hiệu.



Hình 2.2. Cảm biến D203s



Hình 2.3. Cấu tạo bên trong của D203s

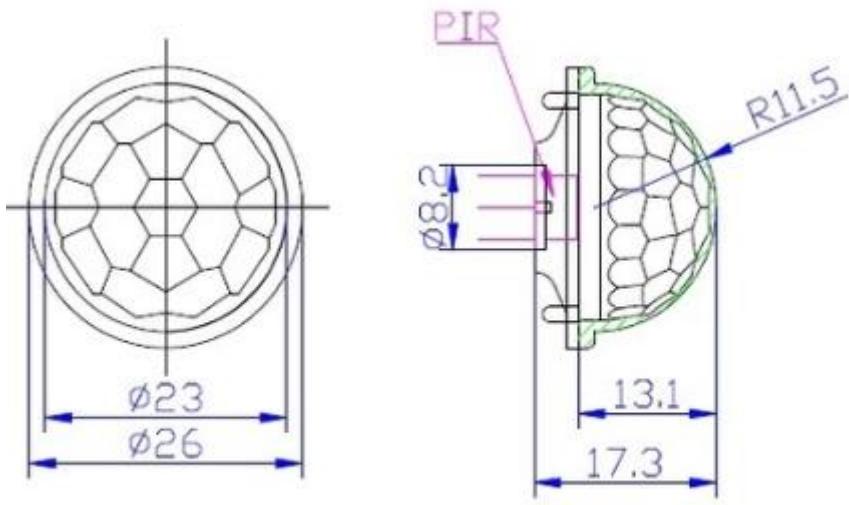


Hình 2.4.Nguyên lý hoạt động

2.1.3 Thấu kính Fresnel hội tụ và tập trung tia hồng ngoại

Thấu kính Fresnel là một thành phần quan trọng trong cấu tạo của cảm biến PIR, đóng vai trò hội tụ và tập trung tia hồng ngoại từ môi trường lên bề mặt chip cảm biến pyroelectric bên trong. Loại thấu kính này được thiết kế đặc biệt để cải thiện hiệu quả thu nhận tín hiệu hồng ngoại từ các nguồn nhiệt khác nhau, đồng thời tăng độ nhạy và mở rộng phạm vi hoạt động của cảm biến.

Thấu kính Fresnel có cấu trúc dạng vòng đồng tâm, bao gồm nhiều lớp mỏng với các góc khúc xạ khác nhau. Thiết kế này giúp giảm đáng kể độ dày và trọng lượng so với các loại thấu kính quang học thông thường, đồng thời duy trì khả năng hội tụ ánh sáng hiệu quả. Khi tia hồng ngoại phát ra từ các nguồn nhiệt như cơ thể người hoặc động vật đi qua phạm vi quan sát của cảm biến, thấu kính Fresnel sẽ tập trung các tia này về trung tâm, tạo ra sự phân bố nhiệt độ rõ ràng trên bề mặt chip cảm biến.



Hình 2.5 Thấu kính Fresnel

Một tính năng quan trọng của thấu kính Fresnel trong cảm biến PIR là khả năng chia nhỏ vùng quan sát thành nhiều vùng phát hiện (detection zones). Mỗi vùng này có thể nhận tia hồng ngoại từ các góc độ khác nhau, tạo ra các tín hiệu nhiệt dao động khi có chuyển động qua lại giữa các vùng. Nhờ vậy, cảm biến không chỉ phát hiện được sự hiện diện mà còn nhận diện được chuyển động, hướng di chuyển, và tốc độ của đối tượng.

Ngoài ra, thấu kính Fresnel còn giúp tăng cường khả năng lọc và tập trung tín hiệu hồng ngoại, loại bỏ nhiễu từ môi trường xung quanh như ánh sáng mặt trời hoặc các nguồn nhiệt không mong muốn. Điều này đảm bảo rằng chỉ những thay đổi nhiệt độ đáng kể và có tính động, chẳng hạn như chuyển động của con người, mới được cảm biến ghi nhận và chuyển đổi thành tín hiệu điện.

2.1.4 Các yếu tố ảnh hưởng tới độ nhạy của cảm biến PIR

Góc quét

Góc quét là một yếu tố quan trọng quyết định phạm vi quan sát của cảm biến. Cảm biến với góc quét rộng (ví dụ: 90° hoặc 120°) có khả năng phát hiện chuyển động trong một khu vực lớn, nhưng dễ bị ảnh hưởng bởi các chuyển động không mong muốn ở rìa vùng quan sát. Trong khi đó, cảm biến có góc quét hẹp giúp tập trung vào một khu vực cụ thể, tăng độ chính xác nhưng lại giảm phạm vi hoạt động.

Một yếu tố liên quan đến góc quét là ống kính Fresnel, được sử dụng để định hình và tăng cường khả năng thu nhận bức xạ hồng ngoại. Chất lượng và thiết kế của ống kính Fresnel đóng vai trò quan trọng trong việc tối ưu hóa độ nhạy. Nếu ống kính không đạt chuẩn hoặc bị hỏng, hiệu suất phát hiện chuyển động của cảm biến sẽ giảm đáng kể.

Nhiệt độ môi trường

Nhiệt độ môi trường ảnh hưởng đáng kể đến hiệu suất của cảm biến PIR, đặc biệt trong việc phát hiện bức xạ hồng ngoại. Khi nhiệt độ môi trường tăng cao, giàn

bằng nhiệt độ cơ thể con người (khoảng 37°C), sự chênh lệch bức xạ hồng ngoại giữa vật thể và môi trường giảm, khiến cảm biến khó phát hiện chuyển động hơn. Điều này thường xảy ra trong các ngày hè nóng bức hoặc ở những không gian kín có nhiệt độ cao.

Ngược lại, trong môi trường nhiệt độ thấp, sự chênh lệch bức xạ giữa vật thể và môi trường tăng lên, giúp cảm biến hoạt động hiệu quả hơn. Tuy nhiên, nhiệt độ quá thấp có thể gây ra các vấn đề như sương mù hoặc nước đọng trên cảm biến, làm suy giảm độ nhạy và độ tin cậy.

Nhiều từ môi trường

Cảm biến PIR dễ bị ảnh hưởng bởi nhiều từ môi trường xung quanh. Các nguồn nhiễu như ánh sáng mặt trời trực tiếp, đèn hồng ngoại hoặc các thiết bị phát nhiệt (như bếp hoặc lò sưởi) có thể tạo ra tín hiệu sai, làm giảm độ chính xác của cảm biến.

Ngoài ra, các chuyển động không mong muốn như cành cây lay động, vật nuôi đi lại hoặc luồng khí nóng từ hệ thống điều hòa cũng có thể kích hoạt cảm biến, dẫn đến cảnh báo giả. Để giảm thiểu tác động của các yếu tố này, cần sử dụng các thuật toán nhận dạng chuyển động hoặc bộ lọc thông minh để loại bỏ nhiễu.

Nhiều điện từ (EMI)

Cảm biến PIR có thể bị ảnh hưởng bởi nhiều điện từ (EMI) từ các thiết bị điện tử khác trong cùng khu vực. Điều này có thể dẫn đến việc giảm độ nhạy hoặc gây ra các lỗi không mong muốn trong hệ thống. Để hạn chế tác động của EMI, cần thiết kế mạch điện tử và vị trí lắp đặt cảm biến phù hợp.

Khoảng cách phát hiện

Phạm vi hoạt động của cảm biến PIR thường được xác định trước, nằm trong khoảng từ 5–12 mét. Tuy nhiên, hiệu suất phát hiện có thể giảm dần ở khoảng cách xa, do cường độ bức xạ hồng ngoại từ vật thể giảm theo khoảng cách. Điều này yêu cầu người dùng phải xem xét vị trí lắp đặt cảm biến để tối ưu hóa khả năng phát hiện trong khu vực cần giám sát.

2.2 Xử lý tín hiệu từ cảm biến PIR

2.2.1 Khuếch đại tín hiệu yếu từ cảm biến

Trong hệ thống cảm biến PIR, tín hiệu đầu ra từ cảm biến thường rất yếu và cần được khuếch đại trước khi có thể sử dụng được. Tín hiệu này thường nằm trong khoảng millivolt, vì vậy mạch khuếch đại đóng vai trò quan trọng trong việc tăng biên độ tín hiệu lên đủ mức để có thể xử lý tiếp tục trong các bộ vi điều khiển hoặc mạch phân tích. Mạch khuếch đại được sử dụng phải đảm bảo không làm thay đổi hình dạng của tín hiệu, đồng thời tăng cường độ mạnh mẽ của tín hiệu, đủ để đạt ngưỡng ADC của vi điều khiển mà không gặp phải hiện tượng nhiễu hay méo tín hiệu.

Trong hệ thống này, sử dụng **bộ khuếch đại operational amplifier (op-amp)** sẽ giúp khuếch đại tín hiệu PIR mà không làm mất mát thông tin. Mạch khuếch đại

cần được thiết kế để có tỷ lệ khuếch đại thích hợp, vừa đủ để đưa tín hiệu vào phạm vi ADC mà không gây ra sự quá tải, đồng thời phải ổn định khi có sự thay đổi về điều kiện môi trường như nhiệt độ, độ ẩm.

2.2.2 Lọc tín hiệu để loại bỏ nhiễu và tín hiệu không mong muốn

Sau khi tín hiệu từ cảm biến PIR đã được khuếch đại, tín hiệu vẫn có thể chứa nhiễu và các tín hiệu không mong muốn gây ảnh hưởng đến độ chính xác và tính ổn định của hệ thống. Những tín hiệu này có thể đến từ nhiều nguồn khác nhau như tín hiệu tần số cao từ các thiết bị điện tử xung quanh, nhiễu do các nguồn điện không ổn định, hay những biến động từ môi trường như sự thay đổi nhiệt độ, độ ẩm hoặc thậm chí ánh sáng. Do đó, để cải thiện chất lượng tín hiệu và đảm bảo hệ thống có thể hoạt động chính xác trong mọi điều kiện, việc lọc tín hiệu để loại bỏ nhiễu là rất quan trọng. Điều này giúp hệ thống chỉ nhận diện các tín hiệu phản ánh chuyển động thực sự và không bị ảnh hưởng bởi các yếu tố ngoại vi không cần thiết.

Bộ lọc tín hiệu là một công cụ quan trọng trong quá trình này, và có thể được phân thành các loại khác nhau tùy vào mục đích sử dụng. Các bộ lọc thông dụng bao gồm **bộ lọc thông thấp (Low-pass filter)** và **bộ lọc thông cao (High-pass filter)**. Mỗi loại bộ lọc này có vai trò và tác dụng riêng biệt trong việc loại bỏ các tín hiệu nhiễu không mong muốn, giúp tín hiệu sau khi lọc trở nên sạch hơn và sẵn sàng cho các bước xử lý tiếp theo.

Bộ lọc thông thấp (Low-pass filter)

Bộ lọc thông thấp (LPF) là bộ lọc chỉ cho phép tín hiệu có tần số thấp đi qua và loại bỏ các tín hiệu có tần số cao hơn ngưỡng cắt của bộ lọc. Trong hệ thống cảm biến PIR, tín hiệu đầu ra từ cảm biến có thể bao gồm các nhiễu tần số cao, chẳng hạn như nhiễu từ nguồn điện, các thiết bị điện tử trong môi trường xung quanh, hoặc các tín hiệu không liên quan đến chuyển động. Những tín hiệu này có thể làm giảm độ chính xác của phép đo chuyển động, vì vậy chúng cần được loại bỏ.

Bộ lọc thông thấp giúp lọc bỏ các tín hiệu tần số cao không mong muốn, chỉ giữ lại những tín hiệu có tần số thấp hơn một ngưỡng đã được xác định trước, phù hợp với tần số của tín hiệu chuyển động mà cảm biến PIR phát ra. Trong hệ thống, tần số tín hiệu chuyển động thường nằm trong dải tần từ 0 đến 5 Hz (tức là những chuyển động rất chậm hoặc không đổi, không phải là các tín hiệu tần số cao). Các tín hiệu nhiễu tần số cao sẽ bị cắt bỏ, giúp hệ thống tập trung vào những tín hiệu có giá trị thực sự, như những thay đổi trong mức độ ánh sáng hồng ngoại phản ánh sự chuyển động của con người. Ví dụ, các tín hiệu từ các nguồn điện 50/60 Hz có thể được loại bỏ để tránh gây nhiễu cho hệ thống. Điều này đảm bảo rằng hệ thống chỉ phát hiện được những thay đổi thực sự có liên quan đến chuyển động, đồng thời tránh được việc phát hiện nhầm hoặc đáp ứng với các yếu tố ngoại lai không cần thiết.

Các bộ lọc thông thấp có thể được triển khai dưới dạng phần cứng, sử dụng các linh kiện điện tử như điện trở, tụ điện hoặc các op-amp, hoặc có thể được thực hiện qua phần mềm với các thuật toán số, tùy thuộc vào yêu cầu của hệ thống và khả

năng xử lý tín hiệu. Trong một số hệ thống, bộ lọc thông thấp có thể có ngưỡng cắt khoảng 10 Hz để đảm bảo tín hiệu chuyển động chậm nhất vẫn có thể được phát hiện rõ ràng.

Bộ lọc thông cao (High-pass filter)

Ngược lại với bộ lọc thông thấp, bộ lọc thông cao (HPF) cho phép tín hiệu có tần số cao hơn một ngưỡng cắt nhất định đi qua, trong khi lọc bỏ các tín hiệu có tần số thấp hơn ngưỡng đó. Trong hệ thống cảm biến PIR, tín hiệu đầu ra có thể có một phần của tín hiệu DC hoặc các nhiễu tần số thấp không liên quan đến chuyển động, chẳng hạn như sự dao động ổn định của môi trường xung quanh hoặc tín hiệu tĩnh. Những tín hiệu này có thể ảnh hưởng đến khả năng phát hiện chuyển động, vì vậy chúng cần được loại bỏ.

Bộ lọc thông cao sẽ loại bỏ các tín hiệu có tần số thấp hoặc tín hiệu DC không cần thiết, chỉ giữ lại những biến động tần số cao phản ánh sự thay đổi nhanh chóng trong môi trường, như sự di chuyển của người hoặc vật thể. Trong hệ thống cảm biến PIR, tín hiệu tĩnh DC có thể là một nguồn nhiễu cần loại bỏ, vì nó không phản ánh sự thay đổi trong môi trường. Một bộ lọc thông cao có ngưỡng cắt khoảng 0.1 Hz có thể loại bỏ được những tín hiệu tĩnh hoặc các dao động chậm không mong muốn, đồng thời giữ lại các tín hiệu nhanh, phản ánh chuyển động thực sự. Điều này giúp hệ thống tập trung vào những thay đổi nhanh, có thể là dấu hiệu của chuyển động thực sự, đồng thời loại bỏ các yếu tố tĩnh hoặc thay đổi chậm mà không liên quan đến chuyển động.

Bộ lọc thông cao có thể được triển khai dưới dạng phần cứng hoặc phần mềm, tương tự như bộ lọc thông thấp, với mục đích giúp làm sạch tín hiệu đầu vào và nâng cao độ chính xác của phép đo chuyển động.

2.2.3 Phân tích tín hiệu ADC

Quá trình phân tích tín hiệu từ cảm biến PIR đóng một vai trò vô cùng quan trọng trong việc phát hiện chuyển động trong các hệ thống Smarthome, giúp nâng cao khả năng giám sát và tự động hóa của ngôi nhà thông minh. Để hệ thống có thể nhận diện và phản ứng chính xác với sự xuất hiện của chuyển động, tín hiệu từ cảm biến PIR cần phải được chuyển đổi và xử lý một cách hiệu quả. Sau khi tín hiệu từ cảm biến đã được khuếch đại và lọc, bước tiếp theo là chuyển đổi tín hiệu analog thành tín hiệu số. Quá trình này là yếu tố quan trọng giúp vi điều khiển hoặc hệ thống xử lý có thể tiếp nhận và phân tích dữ liệu, từ đó đưa ra các quyết định điều khiển hợp lý.

Việc sử dụng bộ chuyển đổi tín hiệu analog sang tín hiệu số (ADC - Analog-to-Digital Converter) trên vi điều khiển ESP32 đóng vai trò quyết định trong việc chuyển đổi tín hiệu từ cảm biến PIR thành dạng dữ liệu có thể xử lý được. Qua đó, tín hiệu analog từ cảm biến PIR được chuyển thành giá trị số để vi điều khiển có thể tiếp nhận và phân tích. Quá trình này bắt đầu với bước lấy mẫu tín hiệu, nơi hệ thống sẽ ghi lại các giá trị của tín hiệu tại các thời điểm cụ thể. Để đảm bảo tín hiệu được lấy mẫu chính xác mà không bị mất thông tin, hệ thống cần phải sử dụng

một tần số lấy mẫu hợp lý. Trong hệ thống này, tần số lấy mẫu được thiết lập là 100Hz, có nghĩa là tín hiệu sẽ được lấy mẫu mỗi 10ms.

Tần số lấy mẫu 100Hz được lựa chọn nhằm đảm bảo rằng tín hiệu từ cảm biến PIR được ghi nhận đủ chi tiết, không bỏ sót các biến động nhỏ, đồng thời vẫn đảm bảo hiệu suất xử lý của hệ thống. Theo định lý Nyquist, để tránh hiện tượng aliasing (sự biến dạng tín hiệu khi tần số lấy mẫu không đủ cao), tần số lấy mẫu phải đủ cao để có thể tái tạo lại đầy đủ đặc tính của tín hiệu gốc. Tần số 100Hz trong trường hợp này là tối ưu, vì nó đảm bảo rằng tất cả các chuyển động mà cảm biến có thể nhận diện đều được ghi lại một cách chính xác.

Khi tín hiệu đã được lấy mẫu, bước tiếp theo là chuyển đổi tín hiệu analog thành tín hiệu số. Vì điều khiển ESP32 sử dụng bộ ADC tích hợp để thực hiện quá trình này. Bộ ADC trên ESP32 có độ phân giải 12 bit, cho phép hệ thống chuyển đổi tín hiệu analog thành các giá trị số với độ chính xác rất cao. Với độ phân giải 12 bit, ADC có thể biểu diễn tín hiệu thành 4096 mức độ khác nhau, giúp hệ thống nhận diện được các biến động rất nhỏ trong tín hiệu, điều này cực kỳ quan trọng đối với cảm biến PIR. Tín hiệu chuyển động từ cảm biến có thể thay đổi rất nhỏ về biên độ, và việc sử dụng ADC với độ phân giải 12 bit giúp phát hiện những thay đổi này một cách chính xác, nâng cao độ nhạy của hệ thống.

Việc sử dụng một ADC có độ phân giải cao còn giúp tăng cường khả năng nhận diện chuyển động trong môi trường có nhiều nhiễu hoặc khi tín hiệu rất yếu. Với các mức độ phân giải thấp hơn, việc phát hiện các biến động nhỏ sẽ trở nên khó khăn hơn, dẫn đến khả năng nhận diện chuyển động bị giảm sút. Vì vậy, độ phân giải 12 bit không chỉ giúp tăng cường độ chính xác mà còn giúp hệ thống hoạt động hiệu quả hơn trong những điều kiện phức tạp.

Khi tín hiệu đã được chuyển thành dạng số, bước tiếp theo là xử lý tín hiệu để phân tích và nhận diện các đặc tính của nó. Một trong những phương pháp phổ biến trong phân tích tín hiệu là phát hiện đỉnh (Peak detection). Phương pháp này giúp nhận diện sự thay đổi đột ngột trong tín hiệu, chẳng hạn như sự chuyển động của người hoặc vật thể trong phạm vi cảm biến PIR. Các thay đổi này thường sẽ thể hiện dưới dạng các đỉnh trong tín hiệu số, và việc phát hiện các đỉnh này là dấu hiệu rõ ràng của sự xuất hiện chuyển động.

Để nâng cao độ chính xác trong quá trình phát hiện chuyển động, các thuật toán xử lý tín hiệu có thể được áp dụng. Một trong số đó là bộ lọc số, có nhiệm vụ làm mượt tín hiệu và loại bỏ các dao động không mong muốn. Các bộ lọc số có thể được thiết kế dưới dạng bộ lọc thông thấp hoặc bộ lọc trung bình để giảm bớt ảnh hưởng của nhiễu, chẳng hạn như tín hiệu tĩnh, sự thay đổi nhiệt độ, hoặc ánh sáng môi trường. Các bộ lọc này giúp tín hiệu đầu ra trở nên sạch sẽ hơn, tăng độ chính xác trong việc phát hiện các biến động nhỏ.

Sau khi tín hiệu đã được xử lý và phân tích, dữ liệu số này sẽ được truyền qua kết nối WiFi của ESP32 lên nền tảng giám sát từ xa, ví dụ như Thingsboard. Việc truyền tải dữ liệu theo thời gian thực không chỉ giúp hệ thống có thể phản ứng nhanh chóng với sự thay đổi trong môi trường, mà còn giúp người dùng có thể giám sát tình trạng của hệ thống Smarthome từ bất kỳ đâu. Dữ liệu này có thể bao

gồm thông tin về sự chuyển động, thời gian phát hiện chuyển động, hoặc các chỉ số khác liên quan đến hiệu suất của hệ thống. Nền tảng Thingsboard cung cấp giao diện trực quan giúp người dùng dễ dàng theo dõi và đánh giá hiệu quả của hệ thống.

Nhờ vào quá trình phân tích tín hiệu ADC, hệ thống Smarthome có thể nhận diện chuyển động một cách chính xác và nhạy bén, từ đó thực hiện các hành động tự động như bật đèn, mở cửa, hay thông báo cho người dùng về sự xuất hiện của chuyển động. Quá trình này không chỉ giúp nâng cao tính hiệu quả của các hệ thống tự động mà còn mang đến một môi trường sống an toàn và tiện nghi hơn cho người sử dụng. Thông qua việc sử dụng các công nghệ như ADC 12 bit và tần số lấy mẫu 100Hz, hệ thống có thể hoạt động ổn định và chính xác, đáp ứng yêu cầu của các ứng dụng Smarthome hiện đại.

2.3 Hệ thống IoT trong Smarthome

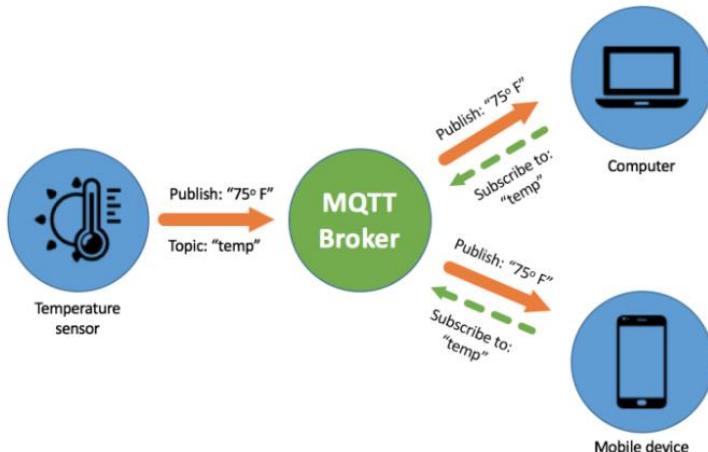
2.3.1 Các giao thức truyền thông phổ biến

Hệ thống IoT (Internet of Things) trong Smarthome là một giải pháp hiện đại giúp tự động hóa và quản lý ngôi nhà một cách thông minh. Thông qua IoT, các thiết bị trong nhà như cảm biến, công tắc, camera an ninh, và các thiết bị gia dụng có thể kết nối và giao tiếp với nhau qua mạng Internet. Điều này cho phép người dùng giám sát, điều khiển thiết bị từ xa và tận dụng các tính năng tự động hóa để tăng cường sự tiện nghi, tiết kiệm năng lượng và đảm bảo an ninh.

Các giao thức truyền thông phổ biến [2]

Để xây dựng một hệ thống IoT trong Smarthome, việc sử dụng giao thức truyền thông phù hợp là rất quan trọng. Dưới đây là hai giao thức phổ biến nhất:

-Giao thức MQTT(Message Queuing Telemetry Transport)

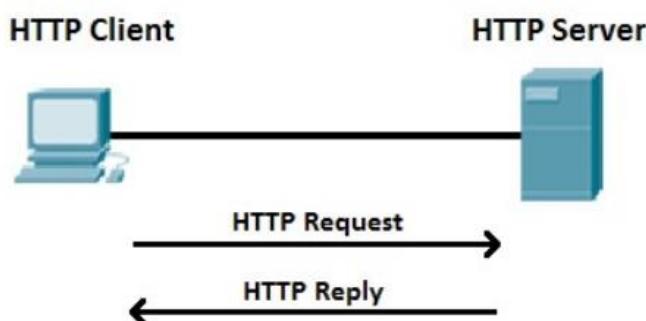


Hình 2.6 Giao thức MQTT

MQTT là một giao thức truyền thông nhẹ được thiết kế đặc biệt cho các thiết bị IoT, tối ưu hóa việc sử dụng tài nguyên hệ thống, hoạt động hiệu quả trong môi trường băng thông hạn chế và duy trì kết nối ổn định ngay cả khi mạng không ổn định. Giao thức này sử dụng mô hình **publish/subscribe**, trong đó các thiết bị gửi dữ liệu (publisher) sẽ truyền thông tin đến một broker trung gian, và các thiết bị khác (subscriber) có thể đăng ký để nhận dữ liệu từ các chủ đề (topics) mà chúng

quan tâm. Với ưu điểm tiêu thụ tài nguyên thấp, khả năng mở rộng cao, và hỗ trợ mã hóa SSL/TLS để bảo mật dữ liệu, MQTT là lựa chọn lý tưởng cho các ứng dụng IoT. Trong hệ thống Smarthome, MQTT được sử dụng để truyền dữ liệu từ các cảm biến như nhiệt độ, độ ẩm, chuyển động hoặc trạng thái thiết bị thông minh về trung tâm điều khiển. Nó cũng hỗ trợ điều khiển từ xa các thiết bị như bật/tắt đèn, điều chỉnh nhiệt độ hoặc theo dõi trạng thái hệ thống qua ứng dụng di động. Với khả năng truyền dữ liệu ổn định và nhanh chóng, MQTT giúp các hệ thống Smarthome hoạt động linh hoạt và hiệu quả hơn.

Giao thức HTTP (Hypertext Transfer Protocol):



Hình 2.7 Giao thức HTTP

HTTP là một giao thức truyền thông phổ biến nhất trên Internet, thường được sử dụng để trao đổi dữ liệu giữa các thiết bị IoT và máy chủ web. Hoạt động theo mô hình **client-server**, các thiết bị IoT đóng vai trò là client, gửi yêu cầu (request) đến máy chủ (server), và nhận lại dữ liệu hoặc lệnh phản hồi. Với tính phổ biến và dễ triển khai, HTTP tích hợp tốt với các ứng dụng web hoặc dịch vụ đám mây, cho phép thiết lập các kết nối IoT một cách đơn giản và nhanh chóng. Trong hệ thống Smarthome, HTTP được ứng dụng để điều khiển thiết bị thông qua giao diện web hoặc ứng dụng di động, chẳng hạn như bật/tắt đèn, kiểm tra trạng thái an ninh hoặc điều chỉnh nhiệt độ. Nó cũng hỗ trợ gửi dữ liệu cảm biến (nhiệt độ, độ ẩm, trạng thái cửa, v.v.) lên server để lưu trữ và phân tích. Mặc dù không tối ưu cho môi trường băng thông hạn chế như MQTT, nhưng với ưu điểm dễ triển khai và khả năng tích hợp rộng rãi, HTTP vẫn là một lựa chọn phổ biến trong các hệ thống IoT và Smarthome.

Tiêu chí	MQTT	HTTP
Kiến trúc	Publish/Subscribe	Client/Server
Kích thước gói tin	Nhỏ (tiêu thụ băng thông thấp)	Lớn hơn, do chứa nhiều thông tin header
Hiệu suất mạng	Hoạt động ổn định trong mạng không liên tục hoặc băng thông thấp	Yêu cầu mạng ổn định, không tối ưu cho băng thông thấp
Cách thức truyền dữ liệu	Gửi dữ liệu qua broker, các subscriber nhận dữ liệu liên quan	Gửi và nhận trực tiếp giữa client và server
Mức tiêu thụ tài nguyên	Thấp	Cao hơn MQTT

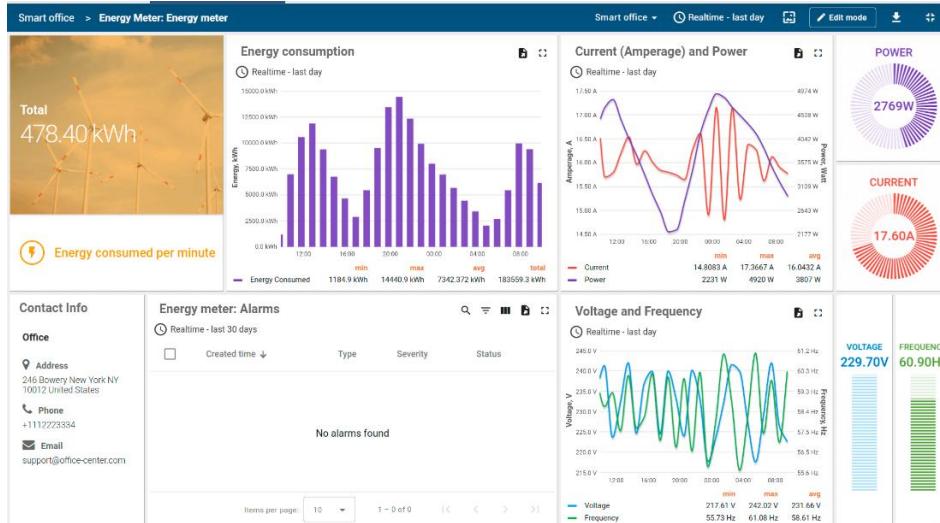
Hỗ trợ thời gian thực	Tốt hơn, nhờ giao tiếp nhanh và nhẹ	Kém hơn, do overhead lớn
Bảo mật	Hỗ trợ mã hóa SSL/TLS	Hỗ trợ mã hóa SSL/TLS
Ứng dụng	Thích hợp cho các thiết bị IoT cần giao tiếp liên tục, dữ liệu nhỏ, môi trường mạng yếu.	Thích hợp cho ứng dụng web, giao tiếp không liên tục và dữ liệu lớn.
Cấu hình và triển khai	Phức tạp hơn một chút, cần cài đặt broker	Dễ triển khai, dựa trên các dịch vụ web phổ biến
Tiêu chuẩn sử dụng	Tối ưu cho IoT, thiết bị cảm biến và hệ thống Smarthome	Phổ biến trong giao tiếp web, ứng dụng di động và dịch vụ đám mây
Ví dụ ứng dụng trong Smarthome	Gửi dữ liệu cảm biến (nhiệt độ, độ ẩm, PIR) và điều khiển thiết bị từ xa.	Điều khiển thiết bị qua giao diện web và lưu trữ dữ liệu cảm biến trên server.

Bảng 1 So sánh MQTT và HTTP

Trong đồ án này, HTTP được chọn vì tính đơn giản, dễ triển khai và tương thích tốt với ThingsBoard qua REST API. HTTP không giới hạn kích thước gói tin như MQTT và không yêu cầu duy trì kết nối liên tục, giúp tiết kiệm tài nguyên hệ thống và đơn giản hóa việc phát triển. Với khối lượng dữ liệu nhỏ như 5 gói tin mỗi giây, mỗi gói 500 byte, HTTP hoàn toàn đáp ứng được yêu cầu mà không gặp phải vấn đề phân mảnh gói tin. Hơn nữa, HTTP cho phép dễ dàng debug và kiểm tra giao tiếp thông qua các công cụ như Postman, giúp mô phỏng và xác minh yêu cầu gửi dữ liệu mà không cần phần cứng.

2.3.2 Thingsboard [3]

Thingsboard là một nền tảng mã nguồn mở dành riêng cho các ứng dụng IoT, cung cấp một bộ công cụ mạnh mẽ để quản lý thiết bị, thu thập và phân tích dữ liệu, cũng như giám sát hoạt động của các hệ thống IoT. Nền tảng này được thiết kế để đáp ứng nhu cầu của những hệ thống Smarthome quy mô lớn, với khả năng xử lý khối lượng dữ liệu khổng lồ và số lượng thiết bị kết nối đa dạng. Nó cung cấp một nền tảng linh hoạt, mở rộng và dễ sử dụng, giúp người dùng có thể dễ dàng triển khai, giám sát và bảo trì các hệ thống Smarthome.



Hình 2.8 Giao diện nền tảng Thingsboard

Hỗ trợ đa giao thức

Một trong những tính năng nổi bật của Thingsboard là khả năng hỗ trợ nhiều giao thức truyền thông phổ biến trong IoT, bao gồm MQTT, HTTP và CoAP. Điều này giúp Thingsboard trở thành một giải pháp lý tưởng cho các hệ thống Smarthome, nơi mà nhiều thiết bị IoT có thể sử dụng các giao thức khác nhau để giao tiếp. MQTT, với đặc tính nhẹ nhàng và khả năng hoạt động ổn định trong môi trường mạng không ổn định, là giao thức phổ biến cho các cảm biến IoT và thiết bị giám sát. HTTP, vốn là giao thức chuẩn trên web, rất phù hợp để kết nối các thiết bị thông minh và dịch vụ đám mây, trong khi CoAP cung cấp khả năng tiết kiệm băng thông và giảm độ trễ cho các ứng dụng yêu cầu phản hồi nhanh. Hỗ trợ đa giao thức giúp các hệ thống Smarthome được xây dựng một cách dễ dàng và linh hoạt, không bị giới hạn bởi yêu cầu của giao thức cụ thể nào.

Quản lý thiết bị

Thingsboard cung cấp một giao diện quản lý thiết bị trực quan và dễ sử dụng. Người dùng có thể dễ dàng thêm mới, cấu hình và theo dõi trạng thái của các thiết bị IoT trong hệ thống Smarthome. Việc theo dõi thiết bị qua nền tảng này giúp người dùng nhận diện nhanh chóng các vấn đề có thể xảy ra, từ đó đưa ra các giải pháp kịp thời để khắc phục sự cố. Giao diện quản lý thiết bị của Thingsboard cho phép người dùng truy cập vào thông tin chi tiết về từng thiết bị, chẳng hạn như trạng thái hoạt động, mức năng lượng tiêu thụ, thông số cảm biến và các sự kiện liên quan. Những thông tin này được cập nhật theo thời gian thực, giúp người quản lý có cái nhìn tổng quan về toàn bộ hệ thống và đưa ra các quyết định hợp lý về việc vận hành và bảo trì thiết bị.

Bảng điều khiển (Dashboard)

Thingsboard cung cấp một công cụ bảng điều khiển mạnh mẽ, cho phép người dùng tạo các dashboard tùy chỉnh để giám sát dữ liệu từ các cảm biến, trạng thái thiết bị và các thông tin khác trong hệ thống Smarthome. Các bảng điều khiển này có thể được thiết kế để hiển thị các chỉ số quan trọng, bao gồm nhiệt độ, độ ẩm, mức độ ánh sáng, chất lượng không khí và nhiều yếu tố khác. Các dữ liệu này có

thể được hiển thị theo thời gian thực, giúp người dùng giám sát trạng thái của các thiết bị và môi trường trong nhà. Ngoài ra, bảng điều khiển còn hỗ trợ hiển thị dữ liệu lịch sử, cho phép người dùng phân tích các xu hướng lâu dài và nhận diện các bất thường. Điều này rất hữu ích trong việc tối ưu hóa hoạt động của hệ thống và thực hiện các điều chỉnh phù hợp để cải thiện hiệu quả năng lượng.

Hệ thống cảnh báo

Một tính năng quan trọng của Thingsboard là hệ thống cảnh báo thời gian thực. Nền tảng này có khả năng tích hợp với các cảm biến và thiết bị giám sát trong hệ thống Smarthome để phát hiện các sự kiện bất thường, chẳng hạn như phát hiện xâm nhập, nhiệt độ vượt ngưỡng an toàn, hoặc mức độ ánh sáng quá thấp. Khi có sự kiện bất thường xảy ra, Thingsboard có thể gửi cảnh báo ngay lập tức đến người dùng qua nhiều kênh khác nhau như email, SMS hoặc thông báo trên ứng dụng di động. Điều này giúp đảm bảo an toàn cho người sử dụng và tài sản trong nhà, đồng thời nâng cao khả năng phản ứng nhanh chóng trước các tình huống khẩn cấp.

Tích hợp dịch vụ đám mây

Thingsboard có khả năng tích hợp mạnh mẽ với các dịch vụ đám mây phổ biến như AWS (Amazon Web Services), Google Cloud và Microsoft Azure. Việc này không chỉ giúp lưu trữ và xử lý dữ liệu từ các thiết bị IoT một cách hiệu quả, mà còn giúp tối ưu hóa khả năng mở rộng của hệ thống. Dữ liệu từ các cảm biến có thể được gửi lên đám mây để phân tích và lưu trữ trong thời gian dài, đồng thời cho phép người dùng truy cập từ bất kỳ đâu và bất kỳ khi nào. Tính năng tích hợp dịch vụ đám mây còn giúp xử lý dữ liệu lớn mà không làm giảm hiệu suất của hệ thống, mở rộng khả năng lưu trữ và phân tích dữ liệu mà không cần đầu tư nhiều vào phần cứng.

Phân tích dữ liệu

Thingsboard cho phép thu thập và phân tích dữ liệu từ các thiết bị IoT để tối ưu hóa hệ thống và nâng cao hiệu quả hoạt động. Dữ liệu thu thập từ cảm biến có thể được phân tích để phát hiện các xu hướng, dự đoán các sự cố tiềm ẩn hoặc tối ưu hóa năng lượng sử dụng trong hệ thống Smarthome. Ví dụ, bằng cách phân tích các dữ liệu về nhiệt độ và độ ẩm, hệ thống có thể tự động điều chỉnh điều hòa không khí hoặc hệ thống sưởi để tiết kiệm năng lượng mà vẫn đảm bảo sự thoải mái cho người sử dụng. Hệ thống phân tích của Thingsboard còn có thể sử dụng các thuật toán học máy (machine learning) để dự đoán các sự cố hoặc vấn đề tiềm ẩn, giúp người quản lý hệ thống phát hiện và xử lý kịp thời.

Bảo mật

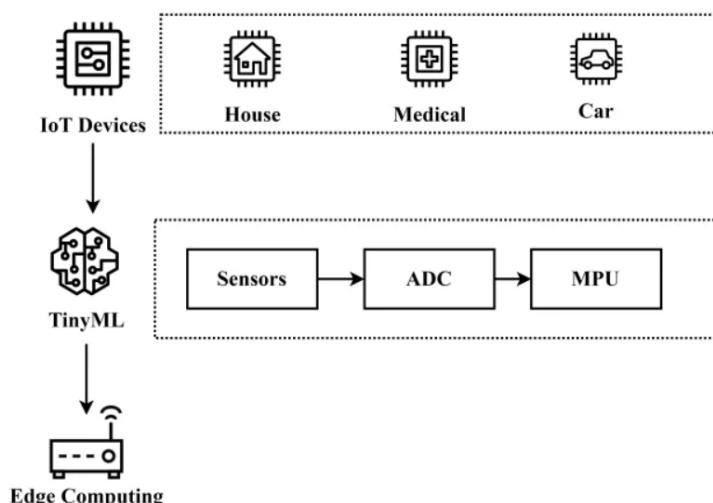
Bảo mật là một yếu tố quan trọng trong bất kỳ hệ thống IoT nào, đặc biệt là trong các hệ thống Smarthome. Thingsboard cung cấp các tính năng bảo mật mạnh mẽ, bao gồm xác thực người dùng, mã hóa dữ liệu và phân quyền truy cập. Điều này giúp bảo vệ hệ thống khỏi các nguy cơ bị xâm nhập hoặc tấn công từ bên ngoài. Xác thực người dùng đảm bảo rằng chỉ những người có quyền truy cập mới có thể điều khiển thiết bị hoặc thay đổi cấu hình hệ thống. Mã hóa dữ liệu giúp bảo vệ thông tin nhạy cảm, chẳng hạn như dữ liệu cảm biến hoặc thông tin cá nhân của

người dùng. Phân quyền truy cập giúp phân chia rõ ràng quyền hạn của các người dùng khác nhau trong hệ thống, từ đó ngăn chặn hành vi xâm phạm hoặc truy cập trái phép vào các chức năng quan trọng.

2.4 Kiến trúc mở và khả năng tích hợp tinyML

2.4.1 Tổng quan về tinyML và ứng dụng trong phân tích tín hiệu cảm biến.

TinyML (Tiny Machine Learning) là một nhánh chuyên biệt của học máy (Machine Learning) được thiết kế để triển khai các mô hình học máy nhẹ, hiệu quả, phù hợp với các thiết bị có tài nguyên hạn chế như vi điều khiển (MCU), cảm biến thông minh, hoặc các thiết bị IoT nhỏ gọn. Điểm đặc biệt của TinyML là khả năng xử lý dữ liệu trực tiếp trên thiết bị "ở biên" (edge), thay vì phải gửi dữ liệu lên các máy chủ đám mây để xử lý. Điều này mang lại một loạt lợi ích đáng kể, bao gồm giảm độ trễ trong phản hồi hệ thống, tiết kiệm băng thông truyền dữ liệu và đảm bảo tính riêng tư của người dùng, do dữ liệu nhạy cảm không phải rời khỏi thiết bị. Với sự phát triển của IoT và nhu cầu ngày càng cao trong việc xử lý dữ liệu thời gian thực, TinyML đang trở thành một công nghệ then chốt trong nhiều ứng dụng.



Hình 2.9 Kiến trúc IoT tích hợp TinyML và Điện toán biên (Edge Computing)

Một trong những lĩnh vực ứng dụng nổi bật của TinyML là phân tích tín hiệu cảm biến. TinyML cho phép các thiết bị nhỏ gọn không chỉ thu thập dữ liệu từ các cảm biến mà còn xử lý và phân tích dữ liệu ngay tại chỗ để đưa ra các quyết định thông minh. Ví dụ, trong nhận dạng mẫu, TinyML có thể phân tích các tín hiệu từ cảm biến chuyển động như IMU (Inertial Measurement Unit) để nhận diện hành vi của con người, như đi bộ, chạy, ngồi hoặc thậm chí phát hiện ngã. Ứng dụng này cực kỳ hữu ích trong các thiết bị chăm sóc sức khỏe, theo dõi người già hoặc hỗ trợ trong thể thao. Ngoài ra, TinyML còn được sử dụng để phát hiện bất thường trong hoạt động của thiết bị công nghiệp hoặc các cảm biến môi trường. Ví dụ, trong một hệ thống giám sát công nghiệp, TinyML có thể theo dõi các rung động hoặc nhiệt độ của máy móc và phát hiện sớm các dấu hiệu bất thường, từ đó ngăn ngừa các sự cố nghiêm trọng hoặc giảm thời gian chết của thiết bị.

Không chỉ dừng lại ở tín hiệu chuyển động hay trạng thái, TinyML còn được ứng dụng mạnh mẽ trong xử lý âm thanh. Một ví dụ phổ biến là nhận diện từ khóa như "OK Google" hoặc "Hey Siri," giúp thiết bị kích hoạt các chức năng chỉ với một câu lệnh ngắn. TinyML cũng có thể phân tích tiếng động, chẳng hạn như nhận biết tiếng vỡ kính trong hệ thống an ninh hoặc nhận diện âm thanh bất thường trong môi trường làm việc, như tiếng rít hoặc tiếng va đập trong nhà máy. Bên cạnh đó, TinyML còn hỗ trợ xử lý hình ảnh đơn giản từ các camera có độ phân giải thấp, như phát hiện chuyển động hoặc nhận diện hình dạng cơ bản. Điều này mở ra cơ hội lớn cho các ứng dụng như giám sát an ninh gia đình, nơi các cảm biến và camera nhỏ gọn có thể hoạt động độc lập mà không cần kết nối liên tục với hệ thống đám mây.

2.4.2 Lợi ích của kiến trúc mở trong các hệ thống TinyML [4]

Kiến trúc mở (Open Architecture) trong các hệ thống TinyML mang lại nhiều lợi ích quan trọng, đặc biệt khi triển khai các thuật toán nâng cao. Đây là một cách tiếp cận thiết kế hệ thống linh hoạt, cho phép mở rộng và tương thích với nhiều nền tảng, phần mềm, và phần cứng khác nhau. Đầu tiên, một trong những lợi ích lớn nhất của kiến trúc mở là khả năng tương thích và tích hợp đa nền tảng. Với thiết kế này, các hệ thống TinyML có thể dễ dàng tích hợp với các giao thức và phần cứng khác nhau, chẳng hạn như các loại cảm biến, vi điều khiển (STM32, ESP32) hoặc các nền tảng phần mềm như TensorFlow Lite for Microcontrollers hay Edge Impulse. Điều này không chỉ giúp các nhà phát triển dễ dàng chuyển đổi giữa các nhà cung cấp phần cứng mà còn cho phép kết hợp nhiều loại cảm biến khác nhau để thu thập dữ liệu đa chiều, ví dụ như tín hiệu nhiệt, chuyển động, âm thanh, và ánh sáng, mà không cần thay đổi toàn bộ hệ thống.

Bên cạnh đó, kiến trúc mở cũng mang lại khả năng mở rộng và tùy chỉnh linh hoạt, cho phép các nhà phát triển dễ dàng thêm các thuật toán nâng cao khi cần mà không cần phải thay đổi cấu trúc cơ bản. Ví dụ, một hệ thống TinyML ban đầu được thiết kế để nhận diện từ khóa có thể được nâng cấp để nhận diện cảm xúc từ giọng nói hoặc phân tích ngữ điệu trong giao tiếp. Ngoài ra, các thư viện ML hoặc framework mới cũng có thể được tích hợp một cách dễ dàng, tận dụng các cải tiến về thuật toán hoặc tối ưu hóa phần cứng, giúp nâng cao hiệu suất hệ thống mà không làm tăng chi phí phát triển.

Một khía cạnh quan trọng khác là kiến trúc mở thúc đẩy sự đổi mới thông qua việc sử dụng mã nguồn mở. Nhiều nền tảng TinyML nổi tiếng như TensorFlow Lite for Microcontrollers, Arduino hoặc Edge Impulse đã tận dụng lợi thế của mã nguồn mở để xây dựng các hệ sinh thái cộng đồng mạnh mẽ. Điều này mang lại nhiều lợi ích, chẳng hạn như khả năng sử dụng lại các thư viện mã nguồn mở từ cộng đồng để tăng tốc độ phát triển, hoặc chia sẻ các thuật toán tối ưu hóa dành cho những bài toán cụ thể như nhận diện âm thanh hay dự đoán tín hiệu thời gian thực. Nhờ đó, các nhà phát triển có thể giảm đáng kể thời gian và công sức cần thiết để xây dựng hệ thống.

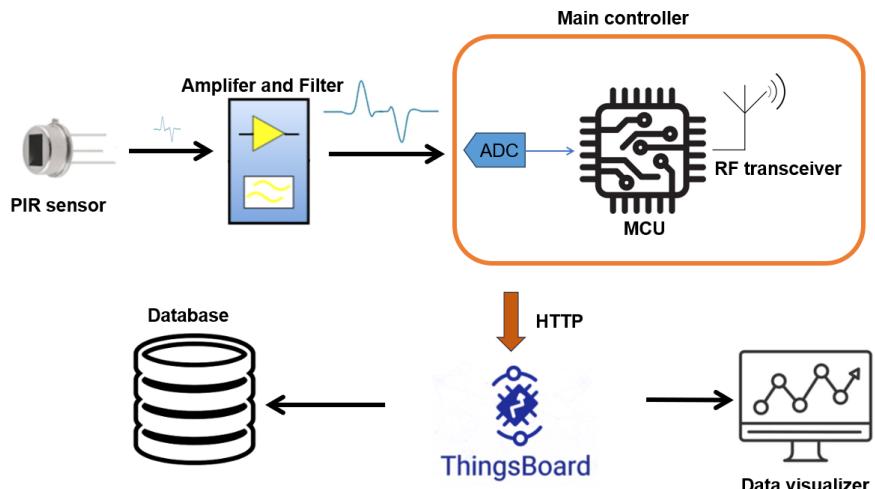
Cuối cùng, kiến trúc mở khuyến khích thiết kế mô-đun hóa, trong đó các chức năng khác nhau của hệ thống, chẳng hạn như xử lý tín hiệu, suy luận TinyML hoặc giao tiếp mạng, được tách biệt thành các mô-đun độc lập. Điều này không chỉ giúp

tối ưu hóa hiệu suất cho từng thành phần riêng lẻ, chẳng hạn như giảm mức tiêu thụ năng lượng trong quá trình xử lý tín hiệu, mà còn cho phép dễ dàng thay thế hoặc nâng cấp từng mô-đun mà không làm gián đoạn hoạt động của toàn bộ hệ thống. Nhờ vậy, kiến trúc mở không chỉ giúp các hệ thống TinyML trở nên linh hoạt và hiệu quả hơn, mà còn mở ra cơ hội cho sự đổi mới và cải tiến liên tục trong lĩnh vực công nghệ này.

CHƯƠNG 3. THIẾT KẾ HỆ THỐNG

3.1 Mô hình hệ thống

Hệ thống được thiết kế để thu thập dữ liệu từ các cảm biến PIR, xử lý dữ liệu trực tiếp tại thiết bị hoặc tại biên, sau đó gửi dữ liệu lên nền tảng Thingsboard. Thingsboard đóng vai trò là server lưu trữ, quản lý và phục vụ các mục đích trực quan hóa cũng như phân tích dữ liệu. Với việc tích hợp Thingsboard, hệ thống có thể tận dụng các tính năng như hiển thị dữ liệu thời gian thực, tạo các bảng điều khiển tùy chỉnh, và thiết lập cảnh báo khi phát hiện sự kiện bất thường, mang lại khả năng quản lý và giám sát dữ liệu một cách hiệu quả.



Hình 3.1 Mô hình hệ thống

Cảm Biến PIR và Đọc Dữ Liệu ADC

Hệ thống sử dụng các cảm biến PIR (Passive Infrared Sensor) để phát hiện chuyển động trong không gian. Các cảm biến PIR này phát hiện sự thay đổi nhiệt độ trong môi trường khi có người hoặc vật di chuyển qua. Tín hiệu đầu ra từ cảm biến PIR là tín hiệu tương tự (analog) với cường độ rất nhỏ. Vì tín hiệu này yếu và có thể bị nhiễu trong quá trình thu nhận, hệ thống sử dụng một mạch lọc và khuếch đại để xử lý tín hiệu.

Mạch lọc giúp loại bỏ các tín hiệu nhiễu không mong muốn, như các sóng điện từ hoặc dao động từ môi trường. Mạch khuếch đại đảm bảo rằng tín hiệu từ cảm biến đạt được mức điện áp phù hợp để được ESP32 xử lý thông qua bộ chuyển đổi tín hiệu từ tương tự sang số (ADC). Các tín hiệu tương tự này sau đó được đưa vào các kênh ADC trên vi điều khiển ESP32 để chuyển đổi thành giá trị số phục vụ cho các bước xử lý tiếp theo.

Đọc và Xử Lý Dữ Liệu trên ESP32

Vì điều khiển ESP32 được cấu hình để thực hiện quá trình đọc dữ liệu từ các cảm biến PIR với tần suất lấy mẫu là 100Hz, nghĩa là hệ thống sẽ thực hiện 100 lần đọc giá trị mỗi giây. Với mỗi lần đọc, tín hiệu đầu ra từ mạch cảm biến (sau khi đã qua lọc và khuếch đại) sẽ được chuyển đổi thành giá trị số bởi bộ ADC.

Sau khi thu thập dữ liệu từ các cảm biến PIR, ESP32 thực hiện xử lý để định dạng dữ liệu. Dữ liệu sau xử lý sẽ được đóng gói thành các gói JSON để chuẩn bị gửi lên server ThingsBoard. Mỗi gói tin JSON bao gồm:

- **Giá trị ADC:** Giá trị số thu được từ cảm biến PIR sau khi chuyển đổi tín hiệu analog.
- **Dấu thời gian (timestamp):** Được lấy từ NTP server (Network Time Protocol) để đồng bộ thời gian chính xác cho từng mẫu dữ liệu.
- **Nhãn vị trí cảm biến:** Mỗi cảm biến được gắn nhãn định danh tương ứng với vị trí của nó, ví dụ: "top", "right", "left", "center", hoặc "bottom".

Nhờ cách đóng gói này, mỗi gói tin JSON chứa đầy đủ thông tin để xác định giá trị cảm biến tại một thời điểm cụ thể và vị trí của cảm biến trong hệ thống.

Gửi Dữ Liệu Lên Server ThingsBoard

ESP32 kết nối với mạng Wi-Fi và sử dụng giao thức HTTP để truyền tải dữ liệu đến server ThingsBoard. Giao thức HTTP được lựa chọn vì tính đơn giản và tương thích cao với nền tảng ThingsBoard.

Việc gửi dữ liệu diễn ra liên tục theo thời gian thực, đảm bảo mọi thay đổi từ các cảm biến PIR được cập nhật ngay lập tức lên server. Dữ liệu được truyền đi dưới dạng các gói JSON, giúp đảm bảo cấu trúc dữ liệu rõ ràng và dễ dàng xử lý trên server.

Lưu Trữ Dữ Liệu Trên Server ThingsBoard

ThingsBoard là một nền tảng IoT mạnh mẽ, không chỉ cung cấp khả năng thu thập và quản lý dữ liệu mà còn lưu trữ dữ liệu một cách an toàn và hiệu quả. ThingsBoard sử dụng cơ sở dữ liệu PostgreSQL làm nền tảng lưu trữ chính.

Dữ liệu từ ESP32 gửi lên sẽ được ThingsBoard ghi vào các bảng trong cơ sở dữ liệu PostgreSQL. Mỗi bản ghi trong cơ sở dữ liệu sẽ bao gồm:

- Giá trị ADC từ cảm biến PIR, biểu thị mức độ phát hiện chuyển động.
- Dấu thời gian, giúp xác định thời điểm dữ liệu được ghi nhận.
- Nhãn vị trí của cảm biến để phân biệt từng cảm biến trong hệ thống.

Cấu trúc cơ sở dữ liệu này cho phép dễ dàng truy vấn và xử lý dữ liệu, phù hợp cho việc phân tích và trực quan hóa sau này.

Phân Tích và Trực Quan Hóa Dữ Liệu

Một trong những tính năng nổi bật của ThingsBoard là khả năng trực quan hóa dữ liệu. Sau khi dữ liệu được lưu trữ trong cơ sở dữ liệu, ThingsBoard cung cấp các công cụ để xây dựng các dashboard với biểu đồ, đồ thị và các bảng dữ liệu chi tiết.

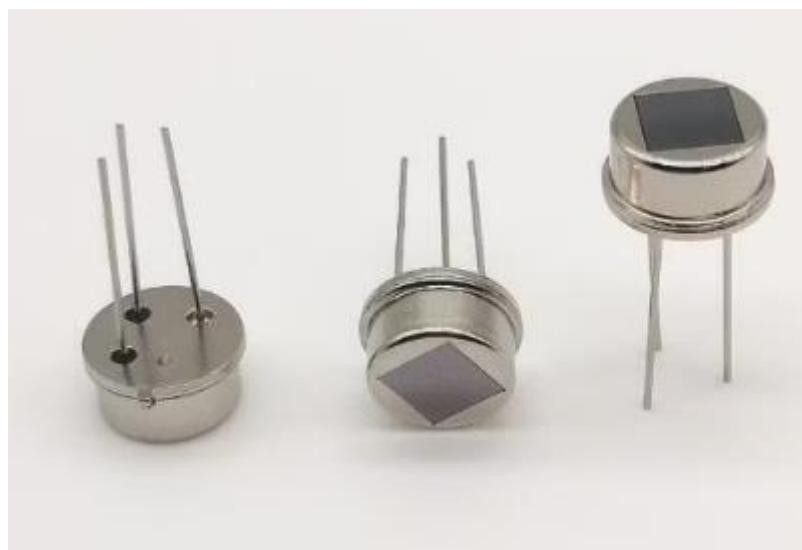
Những công cụ này cho phép người dùng giám sát trạng thái của các cảm biến theo thời gian thực, phân tích các mẫu hành vi và xu hướng trong dữ liệu. Ví dụ, người dùng có thể phát hiện các sự kiện bất thường, theo dõi cường độ chuyển động tại các vị trí cụ thể, hoặc phân tích sự thay đổi theo thời gian để đưa ra các quyết định phù hợp.

ThingsBoard còn hỗ trợ các báo cáo tự động và gửi thông báo qua email hoặc các nền tảng khác khi phát hiện sự kiện đặc biệt. Điều này giúp hệ thống không chỉ hoạt động như một công cụ giám sát mà còn trở thành một hệ thống hỗ trợ ra quyết định thông minh.

3.2 Lựa chọn linh kiện

3.2.1 Cảm biến D203S

Cảm biến PIR D203S là một thiết bị cảm biến hồng ngoại thụ động, được thiết kế để phát hiện sự chuyển động của các vật thể tỏa nhiệt như con người hoặc động vật. Với điện áp hoạt động từ 3V đến 15V, tiết kiệm năng lượng, giúp nó trở thành sự lựa chọn lý tưởng cho các ứng dụng yêu cầu hoạt động liên tục trong thời gian dài mà không tiêu tốn quá nhiều năng lượng. Cảm biến này được ứng dụng rộng rãi trong các hệ thống an ninh, tự động hóa nhà thông minh, và các ứng dụng IoT, giúp phát hiện chuyển động và kích hoạt các thiết bị khác. Độ nhạy cao của cảm biến D203S giúp phát hiện chuyển động chính xác và nhanh chóng, làm cho nó trở thành một lựa chọn phổ biến trong nhiều ứng dụng yêu cầu phát hiện chuyển động hiệu quả và tiết kiệm năng lượng. [1]



Hình 3.2 Cảm biến D203S

3.2.2 IC LM358

LM358 là một op-amp linh hoạt, thích hợp cho cả các ứng dụng điện áp AC thấp và các ứng dụng điện áp DC trung bình. LM358 có thể được tìm thấy dưới dạng gói DIP tiêu chuẩn, vốn quen thuộc với nhiều kỹ sư điện, và cũng có sẵn ở dạng gắn bệ mặt như gói SOIC hoặc SOP, giúp các nhà thiết kế có nhiều tùy chọn về hình thức cho các hệ thống analog và DC. [5]

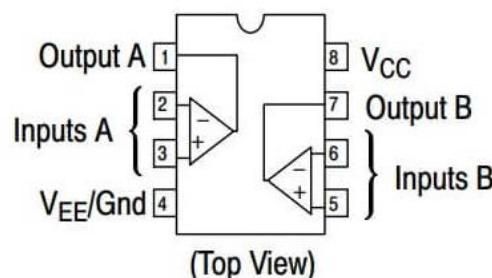
Một đặc điểm quan trọng của LM358 là khả năng hoạt động trên nhiều biến thể từ các nhà sản xuất khác nhau với các thông số kỹ thuật rất giống nhau, đồng thời có kích thước và cấu hình chân phù hợp để dễ dàng thay thế qua lại. Đây là các thông số chính của LM358:

- **Độ khuếch đại tối đa:** lên đến 200.000 lần.

- **Sản phẩm băng thông khuếch đại:** 1-1,2 MHz, cho phép xử lý các tín hiệu tần số thấp.
- **Tỷ lệ loại bỏ chế độ chung (CMRR):** lên đến khoảng 65 dB, giúp giảm nhiễu trong tín hiệu chung.
- **Tỷ lệ loại bỏ nhiễu nguồn (PSRR):** lên đến khoảng 60 dB, giúp ổn định tín hiệu trước nhiễu từ nguồn cấp.
- **Điện áp cung cấp:** linh hoạt từ khoảng 2,5 V đến 30 V, phù hợp cho các ứng dụng sử dụng điện áp thấp hoặc trung bình.
- **Dòng ngắn mạch đầu ra:** khoảng 100 mA, đảm bảo khả năng cung cấp dòng cho các tải nhỏ.

LM358 nổi bật với mức tiêu thụ năng lượng thấp, khả năng hoạt động tốt với điện áp đơn, và tương thích tốt với các hệ thống analog tiêu chuẩn. Với hai kênh khuếch đại tích hợp, LM358 là lựa chọn phù hợp cho các ứng dụng như bộ lọc, mạch so sánh, và bộ khuếch đại tín hiệu nhỏ trong nhiều hệ thống đo lường và điều khiển.

PIN CONNECTIONS



Hình 3.3 Pin connections LM358

3.2.3 MCU ESP32

Tổng quan: ESP32 là một vi điều khiển giá rẻ, năng lượng thấp, nổi bật nhờ khả năng tích hợp WiFi và dual-mode Bluetooth (BLE và Classic Bluetooth). Đây là sản phẩm kế thừa của ESP8266, được phát triển bởi **Espressif Systems**, một công ty có trụ sở tại Thượng Hải, Trung Quốc. ESP32 được sản xuất bởi TSMC bằng công nghệ 40nm, mang lại hiệu suất cao và tiêu thụ năng lượng thấp. [6]

ESP32 sử dụng bộ vi xử lý **Tensilica Xtensa LX6**, hỗ trợ cả biến thể lõi đơn và lõi kép, với nhiều tính năng vượt trội. Vi điều khiển này bao gồm các thành phần như:

- Công tắc antenna tích hợp.
- RF balun.
- Bộ khuếch đại công suất (PA).
- Bộ khuếch đại thu nhiễu thấp (LNA).
- Bộ lọc tín hiệu.
- Module quản lý năng lượng.

Cấu hình phần cứng

- **CPU:** Xtensa Dual-Core LX6 (hệ 32-bit).
 - Tốc độ xử lý: 160 MHz - 240 MHz.
 - Tốc độ xung nhịp đọc Flash: 40 MHz - 80 MHz (tùy chỉnh khi lập trình).
- **RAM:** 520 KB SRAM.
 - Bao gồm 8 KB RAM RTC tốc độ cao và 8 KB RAM RTC tốc độ thấp (dùng cho chế độ Deep Sleep).
- **Bộ nhớ Flash:** Tùy phiên bản, từ 4 MB trở lên.

Giao tiếp không dây

- **WiFi:** Chuẩn IEEE 802.11 b/g/n.
- **Bluetooth:** Dual-mode (BLE và Classic Bluetooth)

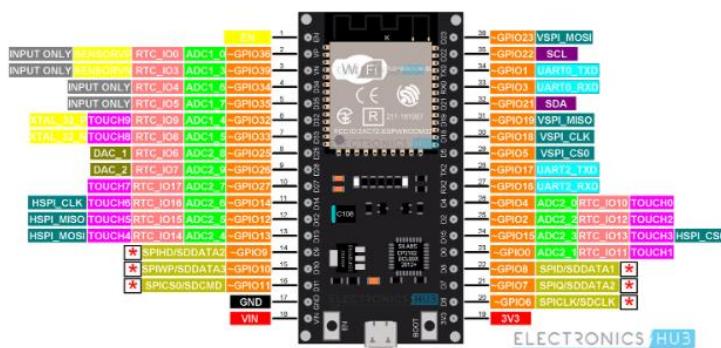
Các giao thức và khả năng ngoại vi:

ESP32 hỗ trợ nhiều giao thức giao tiếp, bao gồm:

- **ADC:** 12-bit, 16 kênh.
- **DAC:** 2 kênh 8-bit.
- **I²C:** 2 cổng.
- **UART:** 3 cổng.
- **SPI:** 3 cổng (1 cổng dành cho chip Flash).
- **I²S:** 2 cổng (hỗ trợ xử lý âm thanh).
- **PWM:** Hỗ trợ trên tất cả các chân GPIO.
- **Ethernet MAC:** Hỗ trợ DMA và chuẩn IEEE 1588.
- **CAN Bus:** Chuẩn CAN 2.0.

Nhiệt độ và điện áp hoạt động

- **Nhiệt độ:** -40°C đến +85°C.
- **Điện áp:** 2.2V đến 3.6V.



Hình 3.4. Modul ESP32

Ứng dụng của ESP32 trong đồ án:

PIR là một cảm biến chuyển động hồng ngoại, được sử dụng để phát hiện chuyển động trong môi trường. ESP32 với khả năng đọc tín hiệu analog thông qua các kênh ADC, là vi điều khiển lý tưởng để xử lý dữ liệu từ PIR Array.

Trong dự án này, chúng ta sẽ sử dụng 5 chân GPIO:

- GPIO32, GPIO33, GPIO34, GPIO35, GPIO39.

Tất cả các chân này đều thuộc ADC1, giúp tránh xung đột với WiFi hoặc Bluetooth khi cần sử dụng các tính năng mạng.

3.2.4 Mô phỏng tín hiệu từ cảm biến D203S

Tín hiệu khi phát hiện có chuyển động qua vùng phát hiện của cảm biến rất nhỏ(cỡ ~ milivon) rất khó để quan sát bằng mắt thường,nên sử dụng bộ khuếch đại.

Mô phỏng tín hiệu từ PIR trên Ltspice:

-Giả lập tín hiệu đầu vào có dạng sau:

SINE(0.5 1m 3 500m 1u 0 1):

Trong đó:

- **0.5 (VO):** Điện áp offset (điện áp dịch) tính bằng volt. Đây là điện áp trung bình của tín hiệu. Tín hiệu hình sin sẽ dao động xung quanh giá trị này. Trong trường hợp này, điện áp trung bình là **0.5V**.

-**1m (VA):** Biên độ điện áp hình sin, tính bằng **volt**. Tín hiệu sẽ dao động lên và xuống từ giá trị offset với biên độ này. Trong trường hợp này, biên độ là **1mV** (0.001V).

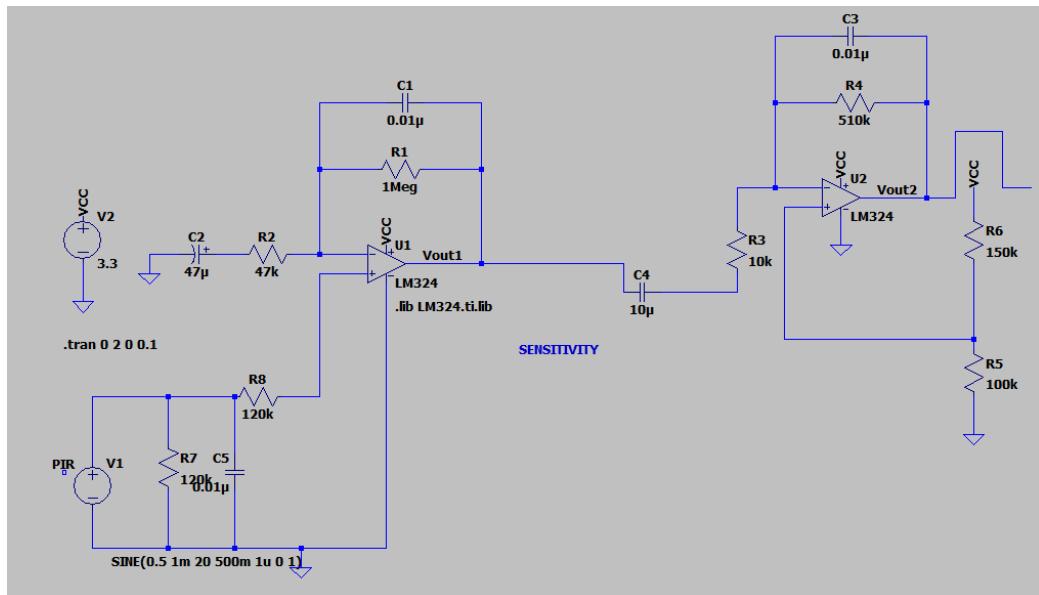
-**3 (FREQ):** Tần số của tín hiệu hình sin, tính bằng **Hz**. Tín hiệu này sẽ dao động với tần số **3Hz** (3 chu kỳ mỗi giây).

-**500m (TD):** Độ trễ (delay), tính bằng **giây**. Đây là thời gian trễ trước khi tín hiệu bắt đầu dao động. Trong trường hợp này, tín hiệu sẽ bắt đầu sau **500ms** (nửa giây).

-**1u (THETA):** Hằng số suy giảm (damping factor), tính bằng **giây**. Giá trị này điều khiển mức suy giảm của biên độ tín hiệu theo thời gian. Với giá trị **1u** (1 micro giây), tín hiệu sẽ suy giảm rất chậm.

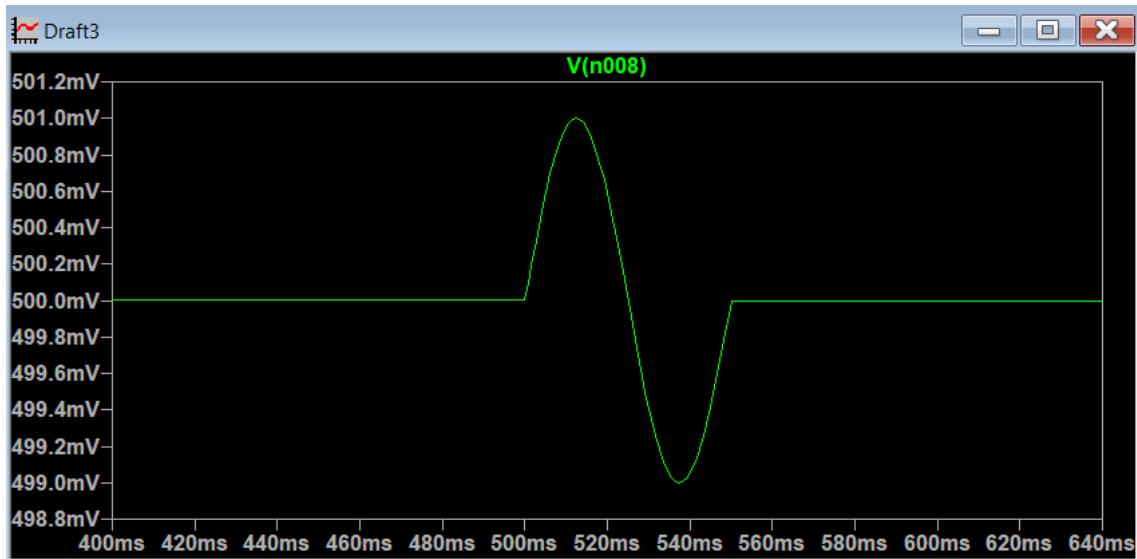
-**0 (PHASE):** Pha ban đầu của tín hiệu hình sin, tính bằng **độ**. Giá trị này quyết định vị trí bắt đầu của tín hiệu trong chu kỳ dao động. Ở đây, tín hiệu bắt đầu từ pha **0 độ** (tại điểm 0 của trực thời gian).

-**1 (N_CYCLES):** Số chu kỳ của tín hiệu. Trong trường hợp này, tín hiệu sẽ tạo ra đúng **1 chu kỳ** rồi dừng lại.



Hình 3.5. Mô phỏng trên LTSpice

Tín hiệu giả lập (trong trường hợp phát hiện có chuyển động qua cảm biến



Hình 3.6. Tín hiệu giả lập

- Giai đoạn kiến trúc đầu tiên khuếch đại tín hiệu. Nó loại bỏ phần DC của tín hiệu và lọc nhiễu tần số cao có thể dẫn đến các phát hiện sai.”

Tần số cao được lọc bởi C4 và R4 với tần số cắt là 15.9 Hz

$$f_{high1} = \frac{1}{2 * \pi * R1 * C1}$$

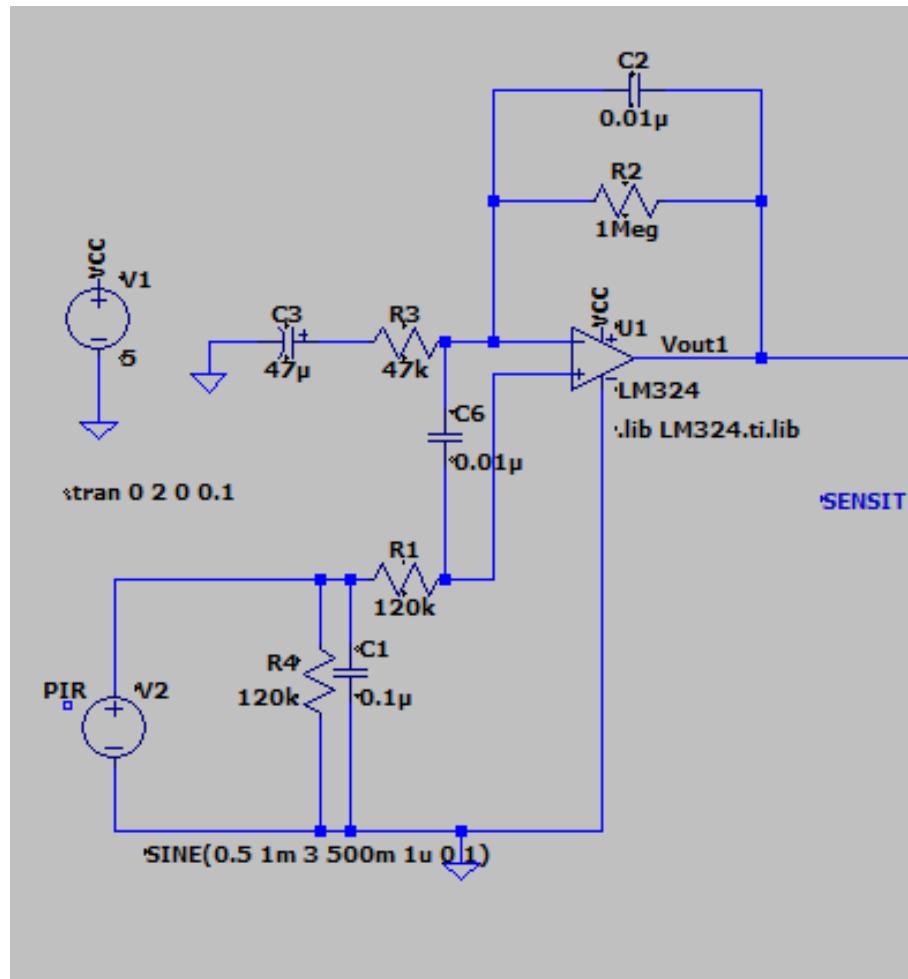
Bộ lọc thứ hai được sử dụng để loại bỏ phần DC của tín hiệu. C3 và R3 tạo thành một bộ lọc thông cao có tần số cắt là 0.072 Hz :

$$f_{low1} = \frac{1}{2 * \pi * R2 * C2}$$

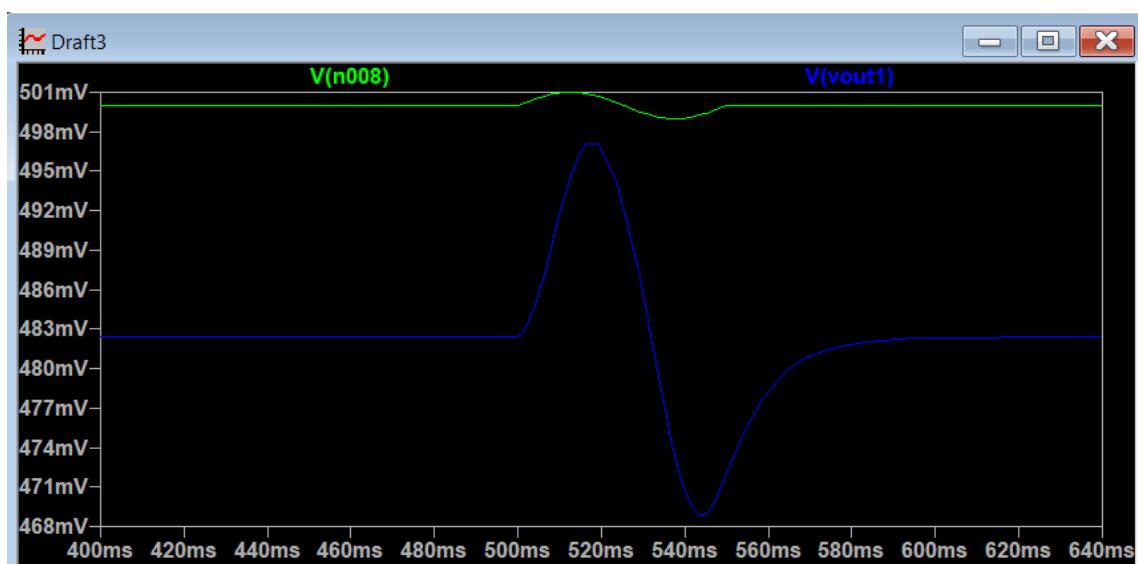
Độ khuếch đại của tầng là 22:

$$G = 1 + \frac{R_1}{R_2}$$

Độ khuếch đại phải đủ cao để khuếch đại tín hiệu cảm biến trên mức nhiễu nhưng không quá cao để đưa bộ khuếch đại hoạt động vào trạng thái bão hòa.



Hình 3.7.Tầng khuếch đại thứ nhất



Hình 3.8.Tín hiệu sau tầng khuếch đại thứ nhất

Giai đoạn khuếch đại thứ 2: Thiết lập độ nhạy của cảm biến PIR

Giai đoạn 2 rất giống với giai đoạn 1. Nó được sử dụng để lọc và khuếch đại tín hiệu AC, ngoại trừ lần này tín hiệu bị đảo ngược.

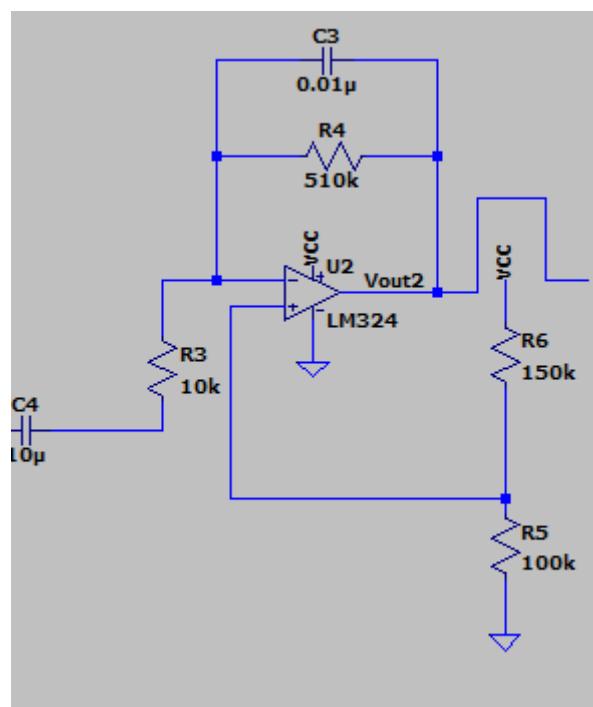
Về chức năng lọc, tần số cắt thấp và cao lần lượt là 1.06 Hz và 31.2 Hz:

$$f_{low2} = \frac{1}{2 * \pi * R3 * C4}$$

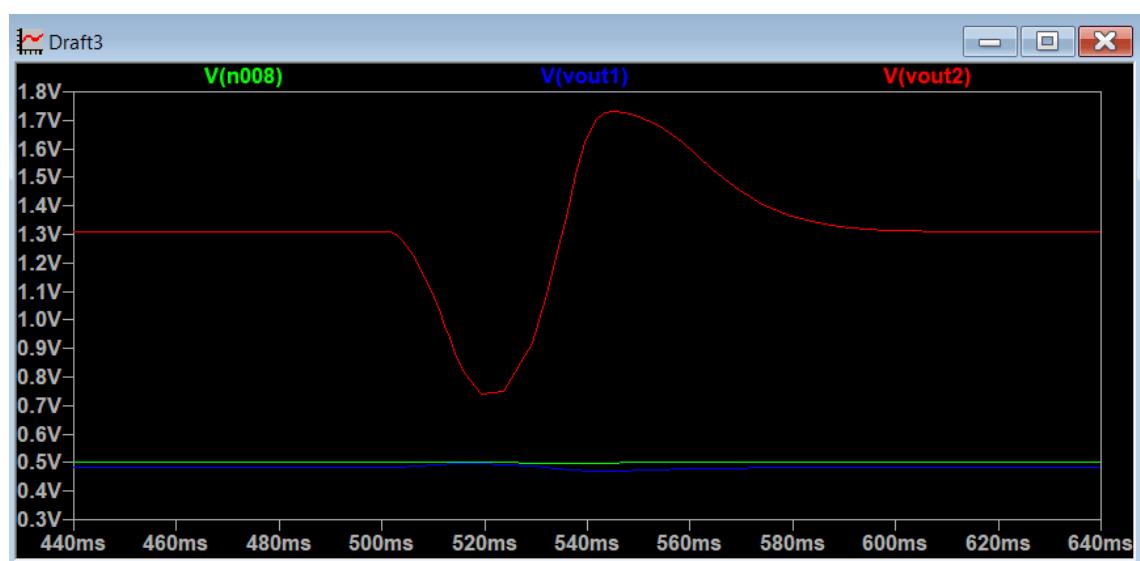
$$f_{high2} = \frac{1}{2 * \pi * R4 * C3}$$

Độ khuếch đại của giai đoạn này là -34 (âm biểu thị tín hiệu đảo ngược) ($\text{Độ khuếch đại}_2 = -R4/R3$). Độ khuếch đại này có nghĩa là sau giai đoạn 2, tín hiệu giữa 1.06 Hz và 31.2 Hz sẽ được khuếch đại 748 lần (57 dB).

Mạch phân áp để điện áp luôn ở mức $0.4*VCC$ (V) khi không có chuyển động

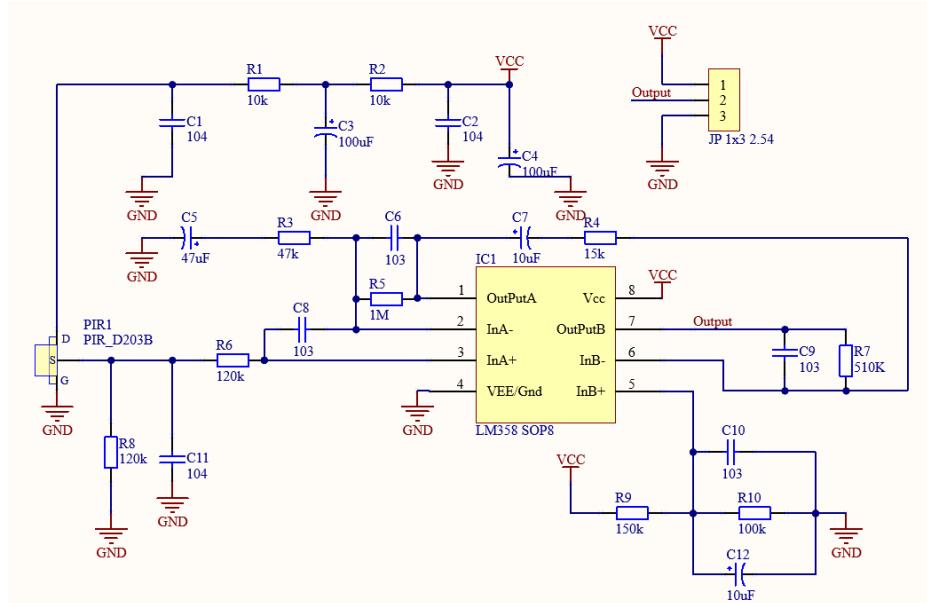


Hình 3.9.Tầng khuếch đại thứ hai

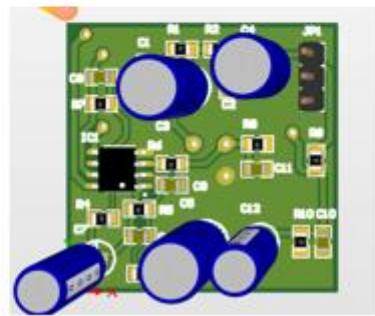
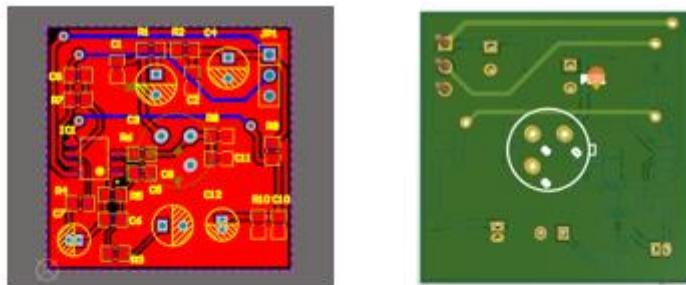


Hình 3.10.Tín hiệu ở tầng khuếch đại thứ hai

3.2.5 Mạch nguyên lý



Hình 3.11.Mạch nguyên lý



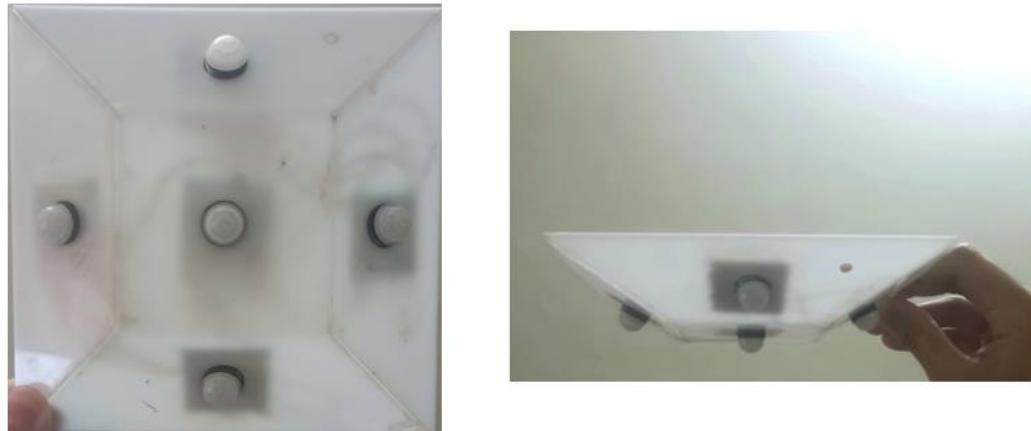
Hình 3.12.Mạch PCB

Kích thước 3x3 cm



Hình 3.13.Mạch sau khi đã hoàn thiện

Các cảm biến được bố trí như sau: 1 cảm biến ở trung tâm và 4 cảm biến ở xung quanh, là các hướng tương ứng là trên, dưới, trái phải, để có thể bao trọn khu vực giám sát, về góc lệch, bố trí sao cho các con xung quanh lệch một góc 30 độ so với phương thẳng đứng



Hình 3.14.PIR Array

3.3 Thiết kế phần mềm

ESP32 sẽ được sử dụng để thực hiện các nhiệm vụ chính bao gồm kết nối Wi-Fi, thu thập dữ liệu từ các cảm biến analog và gửi dữ liệu thu thập được lên nền tảng ThingsBoard. Dữ liệu được gửi sẽ chứa các thông tin quan trọng như thời gian thực (Timestamp), nhãn của từng cảm biến (Label: top, right, bottom, left, center) và giá trị ADC tương ứng được đọc từ các nhãn này. Các nhãn đại diện cho các cảm biến khác nhau, được sử dụng để giám sát các tín hiệu analog từ môi trường hoặc thiết bị ngoại vi. Nhiệm vụ của ESP32 là thực hiện việc lấy mẫu dữ liệu với tần số cao, xử lý dữ liệu và truyền chúng lên máy chủ ThingsBoard để phân tích và hiển thị.

Cụ thể, dữ liệu sẽ được định dạng theo cấu trúc JSON trước khi gửi lên ThingsBoard. Mỗi gói tin dữ liệu sẽ bao gồm các trường như:

- timestamp: Đây là thời gian thực tại thời điểm dữ liệu được thu thập, được lấy từ đồng hồ hệ thống.

-label: Nhãn của cảm biến, cho biết dữ liệu này thuộc về cảm biến nào. Các nhãn sẽ được đặt là "top", "right", "bottom", "left", và "center" tương ứng với 5 kênh ADC được sử dụng.

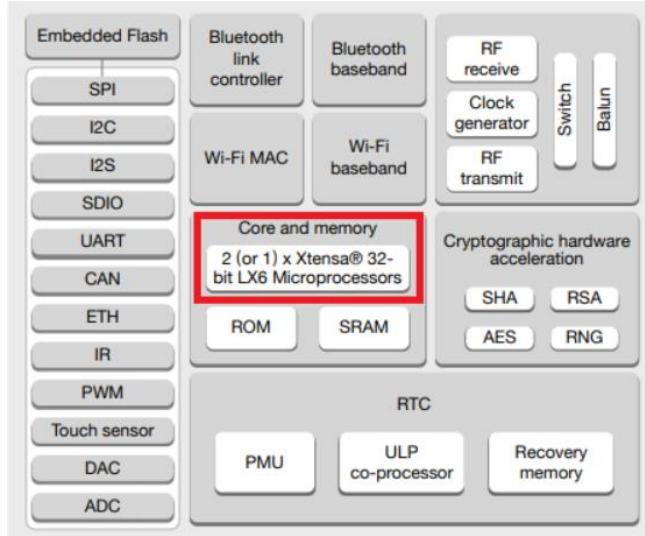
-vols: Đây là danh sách chứa 100 giá trị ADC được thu thập liên tục từ cảm biến tương ứng. Các giá trị được định dạng dưới dạng mảng. Mảng này cho phép ThingsBoard dễ dàng phân tích và xử lý dữ liệu thu thập.

```
{
    "timestamp": 1736614306000,
    "label": "top",
    "vols": [1489, 1488, 1486, 1488, 1488, 1439, 1468, 1477, 1484, 1486, 1486, 1489, 1485, 1485, 1427, 1458, 1473, 1491, 1488, 1483, 1481, 1481, 1486, 1431, 13
97, 1456, 1482, 1435, 1465, 1487, 1491, 1489, 1494, 1463, 1452, 1478, 1479, 148, 1487, 1488, 1491, 1488, 1486, 1487, 1442, 1466, 1498, 1488, 1489, 1494, 1475, 1492, 1488, 1500, 1506, 1506, 1506, 1501, 1503, 1505, 1503, 1509, 1501, 1416, 1382, 1459, 1492, 1501, 1507, 1511, 1509, 1472, 1459, 1488, 1503, 1505, 1507, 1455, 1474, 1491
],
    "label": "right"
},
{
    "timestamp": 1736614306000,
    "label": "bottom",
    "vols": [1487, 1486, 1484, 1487, 1485, 1436, 1469, 1479, 1486, 1488, 1488, 1488, 1486, 1483, 1428, 1457, 1472, 1489, 1477, 1477, 1476, 1479, 1488, 1488, 1444, 14
83, 1454, 1475, 1435, 1465, 1488, 1495, 1495, 1493, 1472, 1459, 1475, 1488, 1493, 1490, 1490, 1490, 1483, 1462, 1471, 1478, 1476, 1479, 1474, 1473, 1479, 1475, 1475, 14
69, 1456, 1472, 1422, 1422, 1450, 1485, 1498, 1501, 1503, 1505, 1503, 1499, 1501, 1416, 1382, 1459, 1492, 1501, 1507, 1501, 1509, 1472, 1459, 1488, 1503, 1505, 1507, 1455, 1474, 1491
],
    "label": "center"
},
{
    "timestamp": 1736614306000,
    "label": "left",
    "vols": [1483, 1479, 1475, 1473, 1468, 1475, 1474, 1475, 1473, 1474, 1469, 1475, 1483, 1462, 1458, 1469, 1462, 1471, 1478, 1476, 1479, 1474, 1473, 1479, 1475, 1475, 147
69, 1472, 1459, 1472, 1475, 1475, 1475, 1475, 1479, 1477, 1481, 1478, 1492, 1485, 1485, 1499, 1487, 1489, 1483, 1483, 1489, 1486, 1484, 1488, 1483, 1477, 1475, 1477, 147
5, 1471, 1473, 1473, 1473, 1478, 1483, 1486, 1487, 1483, 1484, 1478, 1477, 1479, 1471, 1478, 1477, 1478, 1486, 1482, 1479, 1484, 1487, 1487, 1477, 1483, 1482
],
    "label": "center"
},
{
    "timestamp": 1736614306000,
    "label": "center",
    "vols": [1479, 1481, 1482, 1479, 1479, 1478, 1467, 1477, 1476, 1476, 1477, 1479, 1469, 1473, 1475, 1479, 1479, 1481, 1483, 1481, 1479, 1479, 1487, 1487, 1482, 1473, 14
73, 1476, 1481, 1467, 1475, 1476, 1475, 1482, 1479, 1471, 1462, 1483, 1479, 1483, 1475, 1475, 1469, 1466, 1470, 1467, 1476, 1476, 1479, 1477, 1476, 1474, 14
71, 1474, 1472, 1467, 1467, 1469, 1472, 1472, 1474, 1476, 1468, 1473, 1456, 1467, 1476, 1478, 1475, 1478, 1479, 1472, 1472, 1478, 1474, 1475, 1473, 1466, 1467, 1471
],
    "label": "center"
}
}
```

Hình 3.15.Cáu trúc gói tin gửi lên Thingsboard

ESP32 sẽ được cấu hình để sử dụng FreeRTOS với chế độ dual-core, nhằm tối ưu hóa hiệu suất và nâng cao độ tin cậy của hệ thống. Cụ thể, Core 1 sẽ được đảm nhận nhiệm vụ thu thập dữ liệu ADC từ các cảm biến PIR. Core 1 sẽ thực hiện việc lấy mẫu tín hiệu từ các cảm biến, xử lý và chuyển đổi tín hiệu analog thành dữ liệu số. Việc này yêu cầu xử lý liên tục và chính xác, đặc biệt khi có nhiều cảm biến hoạt động đồng thời.

Trong khi đó, Core 2 sẽ chịu trách nhiệm truyền tải dữ liệu đã được xử lý từ Core 1 lên máy chủ ThingsBoard qua giao thức HTTP. Core 2 sẽ đảm nhận việc đóng gói dữ liệu thành các gói JSON, thêm vào dấu thời gian (timestamp), và đảm bảo việc truyền tải dữ liệu liên tục, không bị gián đoạn. Việc phân chia nhiệm vụ này giữa hai lõi giúp tối ưu hóa hiệu suất tổng thể của hệ thống, tránh xung đột trong quá trình xử lý dữ liệu và đảm bảo rằng việc thu thập và truyền tải dữ liệu diễn ra một cách mượt mà và hiệu quả. Điều này không chỉ giúp hệ thống hoạt động nhanh chóng mà còn giảm thiểu độ trễ và tối ưu hóa tài nguyên hệ thống.



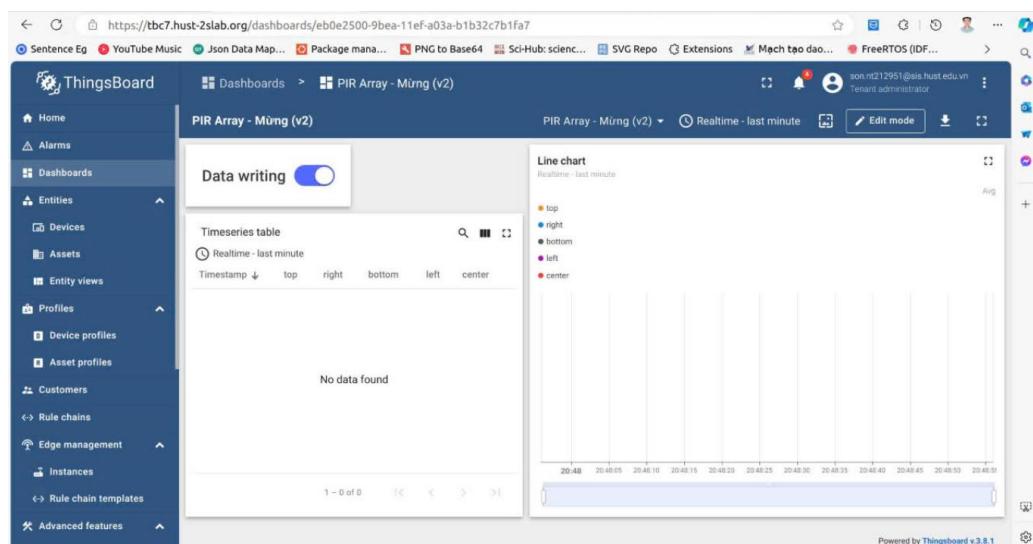
Hình 3.16 Dual-Core trên ESP32

Trên nền tảng ThingsBoard, dữ liệu từ các cảm biến PIR do ESP32 thu thập được hiển thị trực quan thông qua **biểu đồ đường (Chart Plot)** và **bảng dữ liệu (Table)** trên Dashboard.

Biểu đồ đường trực quan hóa giá trị ADC theo thời gian thực, giúp người dùng dễ dàng nhận biết chuyển động trong vùng giám sát. Mỗi cảm biến được biểu diễn bằng một đường đồ thị với màu sắc riêng, nhãn "top", "right", "bottom", "left", "center", giúp theo dõi các thay đổi tín hiệu. Trục X biểu diễn thời gian (timestamp), trục Y biểu diễn giá trị ADC từ 0-4095. Khi có chuyển động, giá trị ADC tạo ra một đường hình sin rõ ràng trên đồ thị, giúp xác định thời điểm phát hiện chuyển động. Bảng dữ liệu gồm các trường:

- **Label:** Nhãn cảm biến.
- **Value:** Giá trị ADC.
- **Timestamp:** Thời gian thu thập dữ liệu

Dữ liệu từ ESP32 được gửi lên ThingsBoard dưới dạng JSON, bao gồm timestamp, label, và chuỗi các giá trị ADC. Sự kết hợp giữa biểu đồ và bảng giúp người dùng vừa theo dõi thời gian thực vừa phân tích chi tiết, đảm bảo hệ thống giám sát hoạt động ổn định và hiệu quả

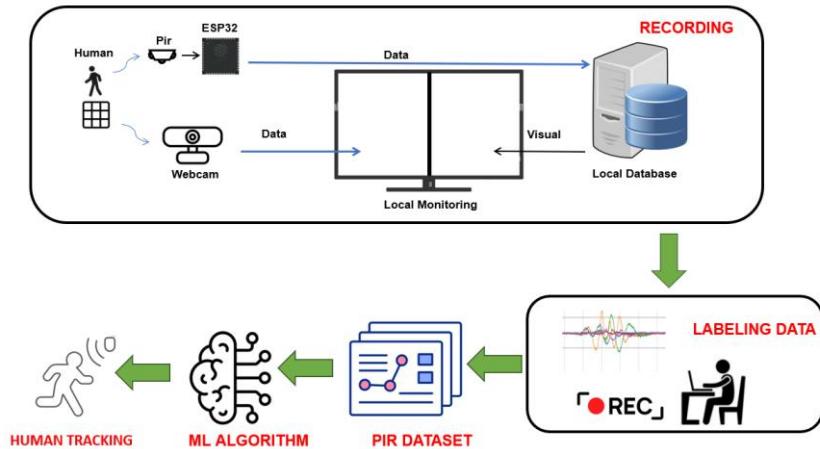


Hình 3.17 Giao diện Thingsboard trực quan hóa dữ liệu

Việc kết hợp biểu đồ đường và bảng dữ liệu trong Dashboard trên ThingsBoard giúp người dùng không chỉ theo dõi tín hiệu từ cảm biến mà còn kiểm soát, điều chỉnh các thiết bị ngoại vi, mang lại trải nghiệm tự động hóa thông minh và hiệu quả trong hệ thống Smarthome

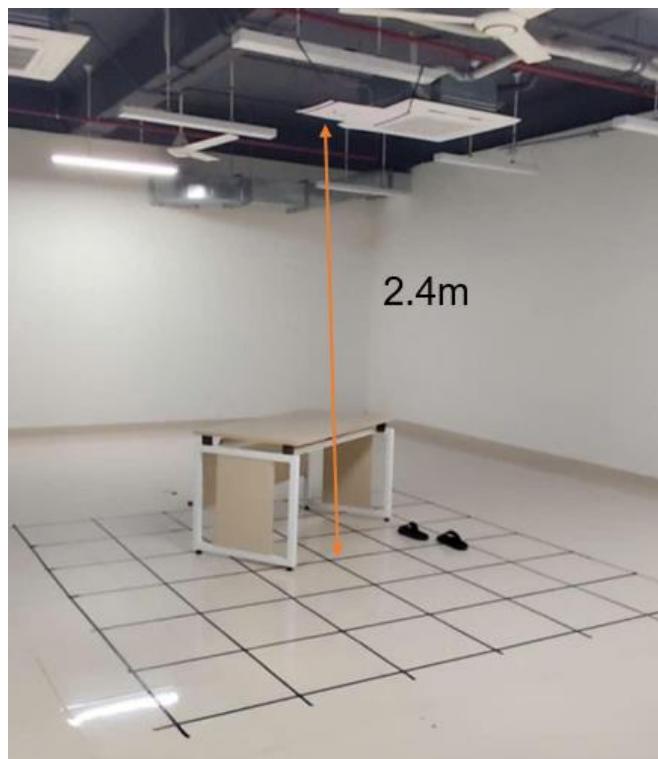
3.4 Kiến trúc mở tích hợp tinyML

3.4.1 Chuẩn bị dữ liệu từ cảm biến để huấn luyện mô hình

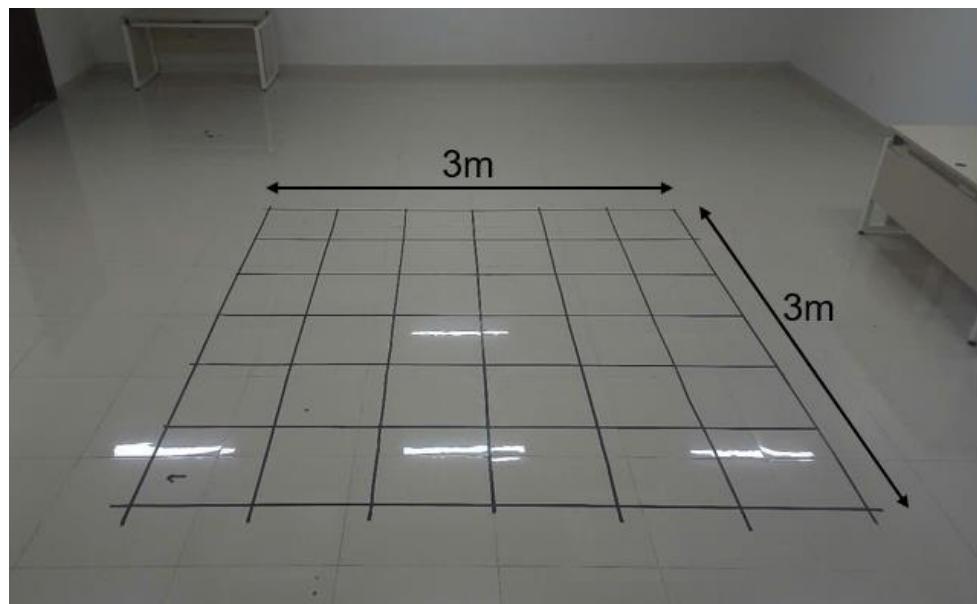


Hình 3.18 Sơ đồ kiến trúc mở tích hợp tinyML

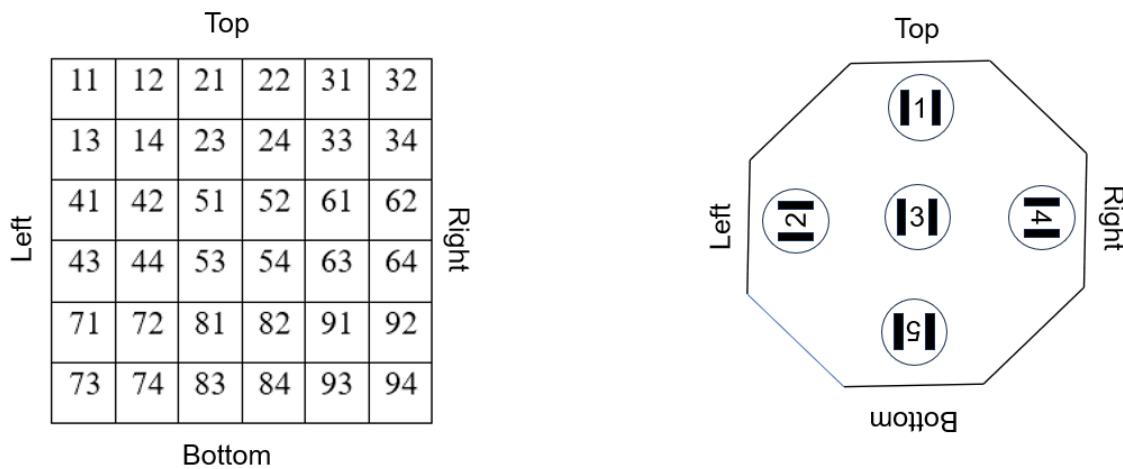
Tiến hành bố trí khu vực có diện tích hình vuông có diện tích 9m²(3mx3m), cảm biến được treo trên trần nhà, chiều cao so với mặt đất là 2.4 m. Khi thu, cần đánh dấu vị trí trên sàn để phục vụ cho quá trình gán nhãn(kích thước vùng thu là 3mx3m chia thành 9 ô 1mx1m, mỗi ô 1mx1m, chia 4 ô nhỏ có kích thước 0.5mx0.5m. Vị trí cảm biến tương ứng với các cạnh của khu vực giám sát và giữ nguyên trong suốt quá trình thu dữ liệu



Hình 3.19. Chiều cao từ nút cảm biến tới mặt sàn



Hình 3.20. Vùng giám sát của cảm biến

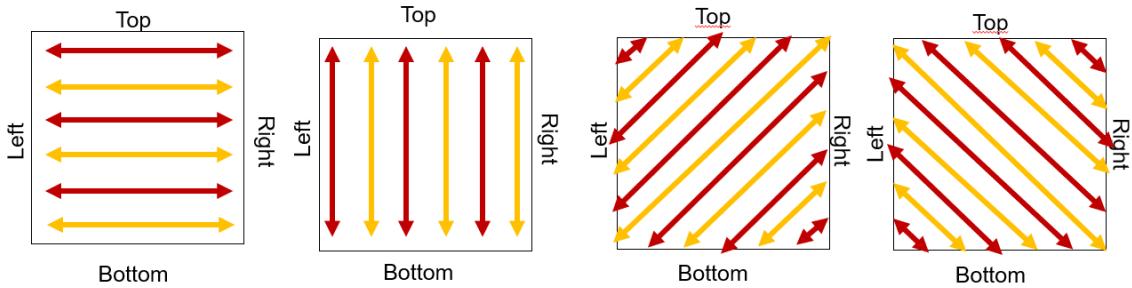


Hình 3.21. Bố trí cảm biến tương ứng các hướng

Khi thu thập dữ liệu với tốc độ mẫu 100 Hz, một yêu cầu quan trọng là người thu thập cần di chuyển đều đặn và ổn định để đảm bảo dữ liệu không bị ngắt quãng hoặc dồn lại. Mọi sự thay đổi đột ngột trong tốc độ di chuyển có thể gây ra sự không đồng đều trong dữ liệu, dẫn đến việc thiếu hoặc trễ dữ liệu. Quá trình thu thập phải liên tục và liền mạch, vì vậy hệ thống cần phải xử lý dữ liệu kịp thời để tránh mất mát và đảm bảo mỗi mẫu dữ liệu phản ánh chính xác trạng thái thực tế.

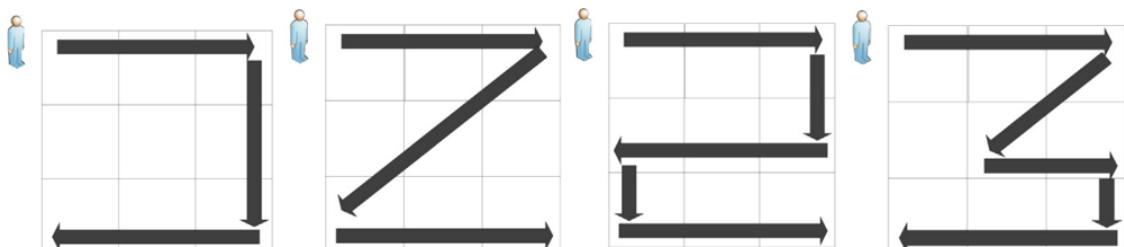
Sau khi thu thập, dữ liệu cần được kiểm tra kỹ lưỡng để đảm bảo tính ổn định và đồng bộ. Việc này giúp phát hiện các lỗi như mất mẫu hoặc trễ thời gian giữa các điểm dữ liệu. Nếu dữ liệu không đồng nhất, cần có các biện pháp xử lý để duy trì sự liên tục và tính chính xác. Quan trọng hơn, dữ liệu cần phải đồng bộ với video thu được để gán nhãn vị trí chính xác. Đồng bộ hóa này tạo điều kiện cho việc phân tích mối quan hệ giữa hành động và các thay đổi trong cảm biến, giúp quá trình phân tích video trở nên hiệu quả và chính xác.

Thực hiện thu theo đường đi kèm video.các mẫu đường đi như sau:



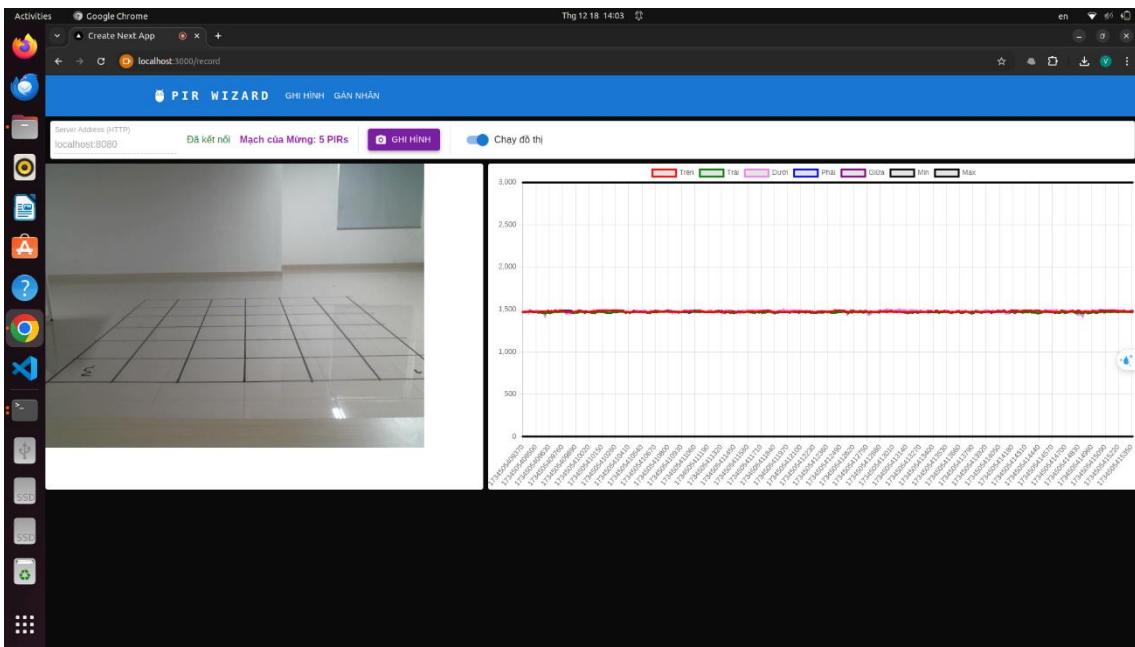
Hình 3.22. Quỹ đạo thu data [7]

Sau khi quá trình thu thập các mẫu dữ liệu hoàn tất, bước tiếp theo là tiến hành thu thập các bộ dữ liệu kiểm tra (testing) phục vụ cho việc đánh giá và huấn luyện các mô hình AI. Các bộ dữ liệu này sẽ được thu thập theo các quỹ đạo cụ thể, được lựa chọn dựa trên mục tiêu và yêu cầu của hệ thống AI. Những quỹ đạo này sẽ mô phỏng các hành động hoặc chuyển động trong các tình huống thực tế mà hệ thống có thể gặp phải, giúp tạo ra một tập dữ liệu phong phú và đa dạng. Quá trình này đóng vai trò quan trọng trong việc cung cấp những dữ liệu chính xác và đầy đủ, giúp mô hình AI có thể học hỏi và đánh giá hiệu suất trong nhiều điều kiện khác nhau, từ đó nâng cao khả năng dự đoán và nhận diện của hệ thống trong các ứng dụng thực tế.



Hình 3.23. Quỹ đạo testcase [7]

Ở đây, khi thu dữ liệu, chúng em không sử dụng Server là thingsboard, mà sử dụng server C++, và 1 web để quay lại quá trình, đồng thời render đồ thị giá trị các giá trị cảm biến theo thời gian thực. Đồng thời, web này có chế độ gán nhãn phục vụ cho quá trình gán nhãn về sau. Video được quay bằng webcam máy tính. Vì một số lý do nên không thể đưa camera lên trần nhà được

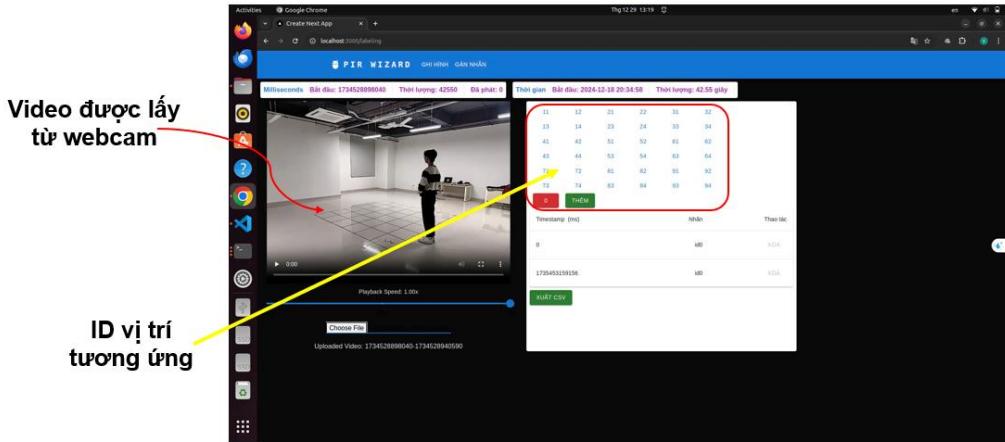


Hình 3.24.Giao diện web quay, thu và trực quan lúc thu

Khi không phát hiện có người trong khu vực giám sát, giá trị ADC của 5 cảm biến PIR ở mức ~1500. Tương đương với điện áp ~1.3V. Khi có chuyển động qua vùng cảm biến nào thì tại cảm biến đó xuất hiện điện áp hình sin. Điện áp này phụ thuộc vào khoảng cách từ người đến cảm biến và diện tích của cơ thể mà cảm biến quét được.

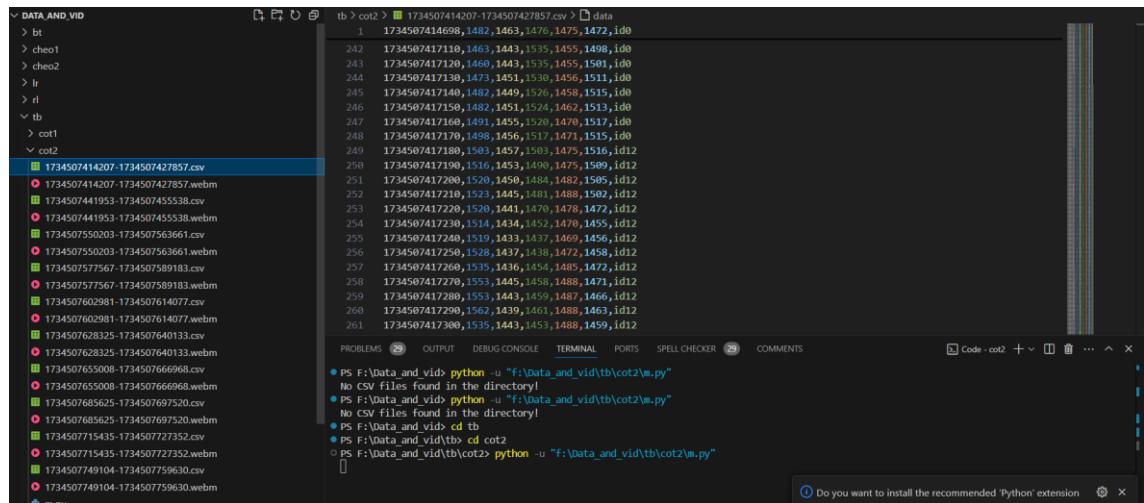
Tiến hành thu trên cột 2 (đi từ 12 đến 74), khi đi từ trên xuống dưới, 10 lần thì thấy rằng, dạng đồ thị của các lần này có nét tương đồng, và khi đi trên các hàng (cột) khác cũng thế, đều có một đặc tính giống nhau, nên có thể sử dụng được đặc điểm này để có thể biết được quỹ đạo của người di chuyển trong khu vực giám sát của cảm biến

Sau khi quá trình thu thập dữ liệu hoàn tất, bước tiếp theo là tiến hành gán nhãn cho các mẫu dữ liệu. Trong trường hợp này, các nhãn cần được gán để chỉ rõ vị trí của người trong khu vực giám sát, không liên quan đến hành động của người đó. Việc gán nhãn này phải được thực hiện chính xác, đảm bảo rằng mỗi nhãn phản ánh đúng vị trí mà người di chuyển trong suốt quá trình thu thập dữ liệu. Điều này có nghĩa là mỗi điểm dữ liệu phải được liên kết với một vị trí không gian cụ thể, đồng bộ với thời gian thu thập, để phản ánh chính xác sự di chuyển của người trong khu vực giám sát. Quá trình gán nhãn phải đảm bảo rằng các vị trí được xác định một cách chính xác và khớp với các điểm thu thập dữ liệu, giúp đảm bảo tính chính xác và độ tin cậy cho mô hình AI trong việc phân tích sự di chuyển của người.



Hình 3.25 Giao diện gán nhãn

Các dữ liệu về điện áp của cảm biến đều được xuất dưới dạng .CSV kèm với vid, phục vụ cho bên AI gán nhãn và kiểm tra tính đúng đắn. Sau đó sẽ đưa ra kết quả cho bài toán này



Hình 3.26. Cấu trúc file.csv

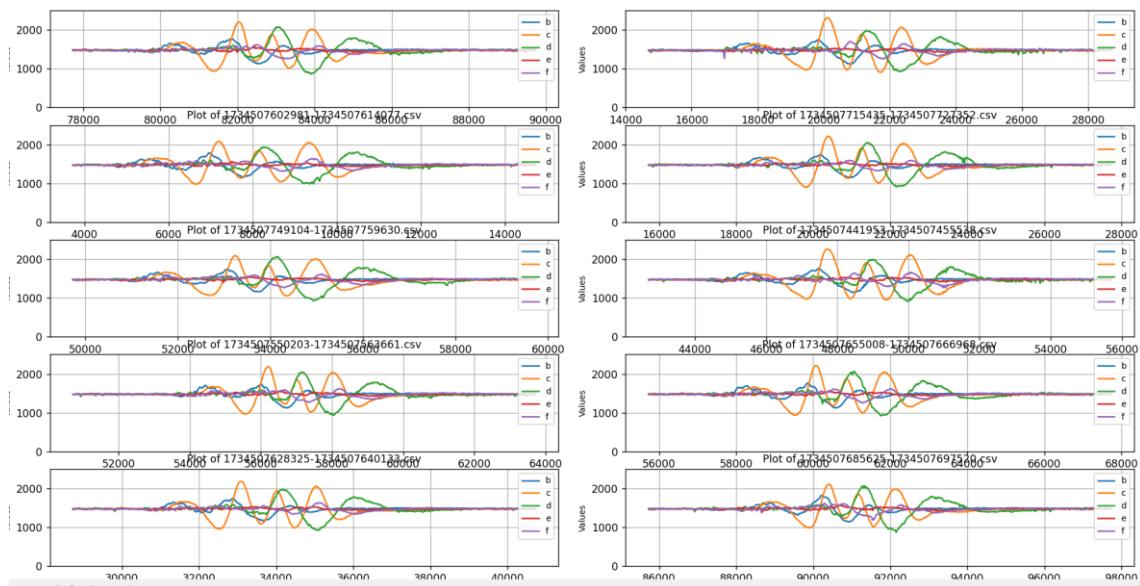
Tên của file dữ liệu .csv và video sẽ được lưu trữ dưới dạng thời gian bắt đầu và thời gian kết thúc của quá trình ghi. Đây là một cách tổ chức dữ liệu hợp lý và dễ dàng theo dõi, giúp người dùng dễ dàng xác định khoảng thời gian mà dữ liệu và video liên quan đến. Ví dụ, tên file có thể bao gồm thông tin như "start_time_end_time.csv" và "start_time_end_time.webm", nơi "start_time" và "end_time" là thời gian bắt đầu và kết thúc của quá trình ghi. Cách đặt tên này không chỉ giúp lưu trữ dữ liệu một cách có hệ thống mà còn giúp dễ dàng liên kết giữa các file video và dữ liệu thu thập trong suốt quá trình giám sát.

Bên trong file .csv, dữ liệu sẽ được tổ chức thành 6 cột. Cột đầu tiên chứa thông tin về thời gian, phản ánh thời điểm chính xác mà các giá trị cảm biến được ghi lại. Các cột tiếp theo, từ cột thứ hai đến cột thứ sáu, sẽ là các giá trị thu được từ các cảm biến khác nhau, phản ánh các thông số mà cảm biến đo được trong khoảng thời gian tương ứng. Cuối cùng, cột cuối cùng trong file .csv chứa thông tin về vị

vị trí của người trong khu vực giám sát tại thời điểm đó. Vị trí này là kết quả của quá trình gán nhãn, nơi mỗi giá trị trong cột này được xác định dựa trên sự di chuyển của người trong khu vực giám sát, đồng bộ với thời gian và dữ liệu thu thập từ các cảm biến.

Cách thức tổ chức này cho phép việc phân tích và xử lý dữ liệu trở nên thuận tiện hơn, đồng thời giúp đảm bảo rằng mỗi bản ghi dữ liệu không chỉ có thông tin về thời gian và các cảm biến mà còn có mối liên kết rõ ràng với vị trí của người, tạo điều kiện thuận lợi cho các mô hình phân tích AI trong việc đánh giá sự di chuyển của người trong không gian giám sát.

Dưới đây là đặc tính khi đi từ ô 12 đến ô số 74



Hình 3.27. Dạng tín hiệu của trường hợp đi ở cột thứ hai

3.4.2 Đề xuất hướng mở rộng để tích hợp tinyML trong tương lai.

Thay vì chỉ báo động khi có chuyển động, hệ thống có thể phân tích dữ liệu từ nhiều cảm biến PIR (Passive Infrared Sensor) để xác định người đang ở đâu và họ di chuyển như thế nào trong không gian. Việc này sẽ giúp nâng cao độ chính xác trong việc nhận diện và theo dõi hành động của người trong không gian giám sát.

Sau khi thu thập đủ dữ liệu từ các cảm biến PIR, bước tiếp theo là huấn luyện mô hình TinyML để phân tích các mẫu chuyển động. Mô hình sẽ được huấn luyện để nhận diện các đặc điểm chuyển động của người. Đặc biệt, mô hình TinyML sẽ được thiết kế để xác định hành động và hướng di chuyển của người dựa trên các thay đổi tín hiệu từ các cảm biến PIR khác nhau. Các thuật toán học máy như phân loại và học sâu có thể được áp dụng để phân tích dữ liệu từ các cảm biến và dự đoán vị trí cũng như hướng đi của người. [8]

Sau khi huấn luyện mô hình, mô hình này sẽ được triển khai trực tiếp trên vi điều khiển ESP32 để xử lý dữ liệu ngay tại chỗ. Điều này có nghĩa là tất cả các dữ liệu từ cảm biến PIR sẽ được thu thập, xử lý và phân tích ngay trên ESP32 mà không cần truyền tải dữ liệu lớn đến các server ngoài. Việc xử lý tại chỗ sẽ giúp giảm tải băng thông, cải thiện hiệu suất và giảm độ trễ trong việc theo dõi chuyển động. Mô hình TinyML sẽ phân tích các tín hiệu chuyển động và đưa ra kết quả về vị trí và

hướng di chuyển của người. Các kết quả này sẽ được gửi đến hệ thống giám sát hoặc ứng dụng của người dùng qua kết nối mạng để phục vụ việc theo dõi thời gian thực.

Sau khi phân tích và xác định vị trí của người, ESP32 sẽ gửi kết quả này đến một server hoặc hệ thống giám sát qua các giao thức mạng như MQTT hoặc HTTP. Tại đây, các dữ liệu sẽ được hiển thị trên một giao diện người dùng (UI) để người quản lý có thể theo dõi và phân tích tình hình. Dữ liệu có thể bao gồm thông tin về vị trí của người trong khu vực giám sát, hành động tiếp theo của họ, và thậm chí là lịch sử di chuyển.

CHƯƠNG 4. THỰC NGHIỆM ĐÁNH GIÁ

4.1 Thủ nghiệm phần cứng

Theo tính toán, trong trạng thái không phát hiện chuyển động, tín hiệu đầu ra của cảm biến PIR duy trì ổn định ở mức $\sim 1.3V$, tương ứng với giá trị ~ 1500 khi đọc qua ADC. Khi phát hiện chuyển động, điện áp đầu ra sẽ thay đổi, tăng hoặc giảm tùy thuộc vào hướng di chuyển của đối tượng so với cảm biến. Cường độ tín hiệu không chỉ phụ thuộc vào khoảng cách gần hay xa cảm biến, mà nó còn phụ thuộc vào diện tích bề mặt cơ thể mà cảm biến quét được. Nếu bề mặt diện tích quét được rộng thì cường độ tín hiệu càng rõ rệt. Bộ lọc tín hiệu cũng đóng vai trò quan trọng trong việc loại bỏ nhiễu và tín hiệu không mong muốn, đặc biệt trong môi trường có nhiều nguồn gây nhiễu như ánh sáng hoặc các dao động nhiệt. Việc lọc tín hiệu đã đảm bảo rằng các tín hiệu đầu ra không bị ảnh hưởng bởi nhiễu nền, các tín hiệu điện áp không còn hiện tượng sụt áp.



Hình 4.1 Tín hiệu của mảng cảm biến khi chưa phát hiện chuyển động

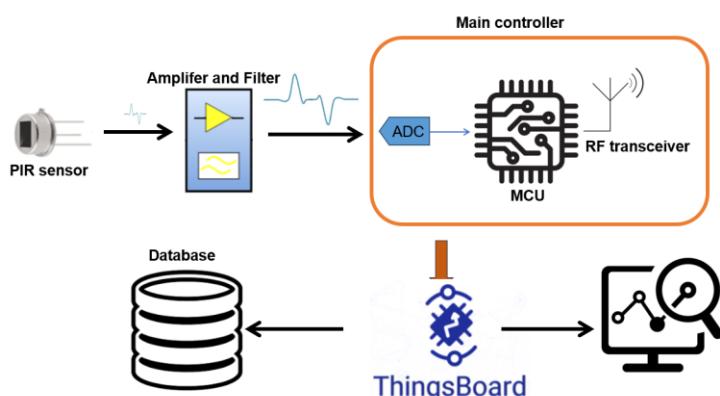


Hình 4.2 Tín hiệu của mảng cảm biến phát hiện có chuyển động trong khu vực giám sát

Sau khi treo mảng cảm biến lên trần nhà ở độ cao 2.8m, nhận thấy cảm biến có khả năng phát hiện chuyển động của con người trong phạm vi khoảng 3m. Mặc dù có thể tăng độ nhạy của cảm biến để mở rộng phạm vi phát hiện, tuy nhiên, điều này có thể dẫn đến hiện tượng bão hòa tín hiệu, đặc biệt khi đối tượng ở quá gần cảm biến. Hiện tượng bão hòa làm điện áp đầu ra vượt ngưỡng, dẫn đến mất dữ liệu, ảnh hưởng đến độ chính xác của bài toán theo dõi vị trí người. Vì vậy, em quyết định giữ độ nhạy của cảm biến ở mức thấp để đảm bảo tính ổn định và chính xác của hệ thống.

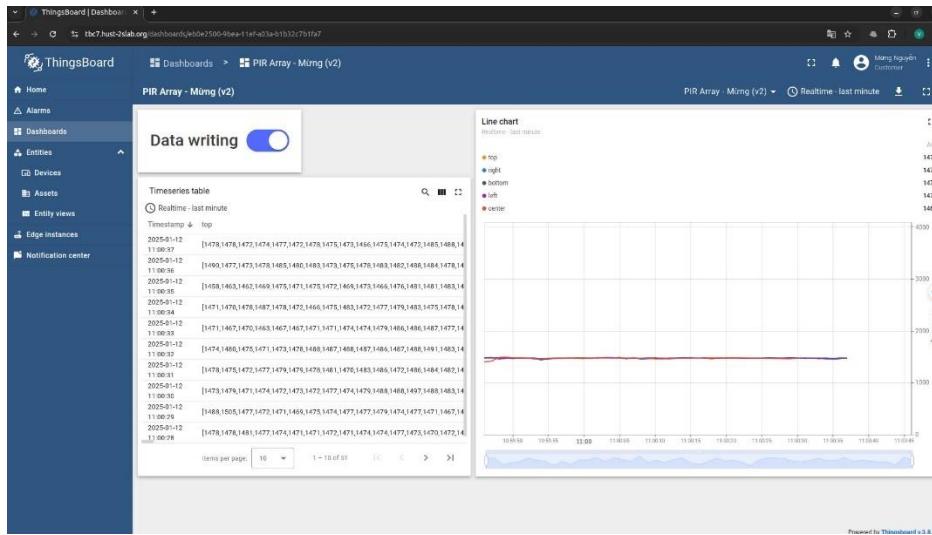
4.2 Thủ nghiệm phần mềm IOT

ESP32 được cấu hình để kết nối với mạng WiFi, sử dụng SSID và mật khẩu thích hợp. Sau đó, hệ thống liên tục đọc giá trị ADC từ cảm biến PIR với tần số 100Hz, và mỗi giây gửi dữ liệu của tất cả 5 cảm biến, mỗi cảm biến chứa 100 giá trị kèm thời gian lấy từ NTP lên nền tảng Thingsboard qua giao thức HTTP. Trên Thingsboard, các giá trị này được trực quan hóa dưới dạng đồ thị, giúp người dùng dễ dàng theo dõi sự thay đổi tín hiệu từ cảm biến theo thời gian.



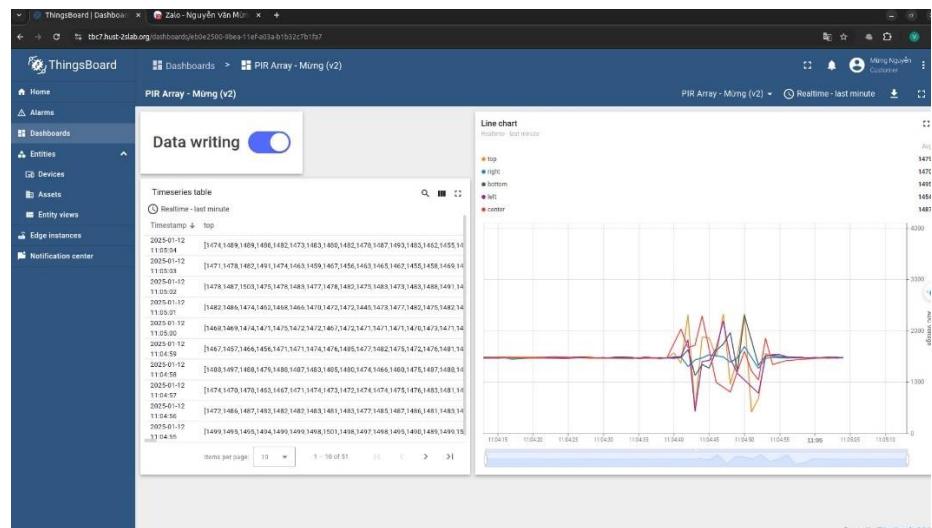
Hình 4.3 Mô hình hệ thống

Để hỗ trợ theo dõi và phân tích dữ liệu, trên Thingsboard, em bố trí một biểu đồ Line chart để thể hiện sự biến động của dữ liệu cảm biến được gửi lên từ ESP32. Đồng thời, một bảng Timeseries table được thiết lập để kiểm tra và đảm bảo không có gói tin nào bị mất. Bảng này hiển thị thời gian nhận được gói tin cùng các giá trị từ cảm biến, giúp theo dõi mọi thay đổi trong quá trình truyền tải.Thêm vào đó, một nút Data Writing được tích hợp, cho phép bật hoặc tắt chức năng thu thập và truyền tải dữ liệu.Khi giá trị của cảm biến nằm trong ngưỡng phát hiện chuyển động,sẽ có thông báo hiện ra trên Dashboard thingsboard

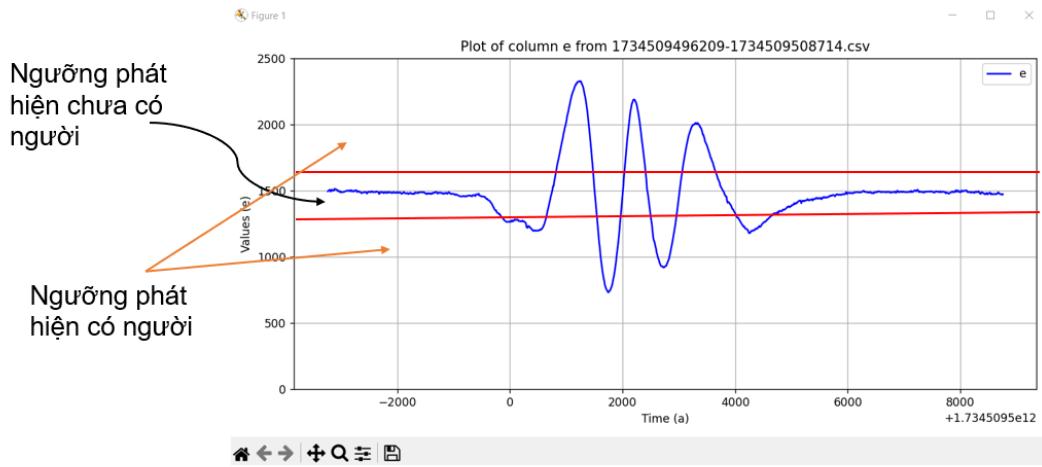


Hình 4.4 Đồ thị trực quan trên Thingsboard khi chưa phát hiện chuyển động

Khi không có người trong phạm vi phát hiện của cảm biến, đồ thị trên Thingsboard sẽ hiển thị các giá trị ổn định. Tuy nhiên, khi phát hiện chuyển động, đồ thị sẽ xuất hiện các đường hình sin, phản ánh sự dao động mạnh mẽ của tín hiệu, tương ứng với sự chuyển động của đối tượng trong phạm vi cảm biến.

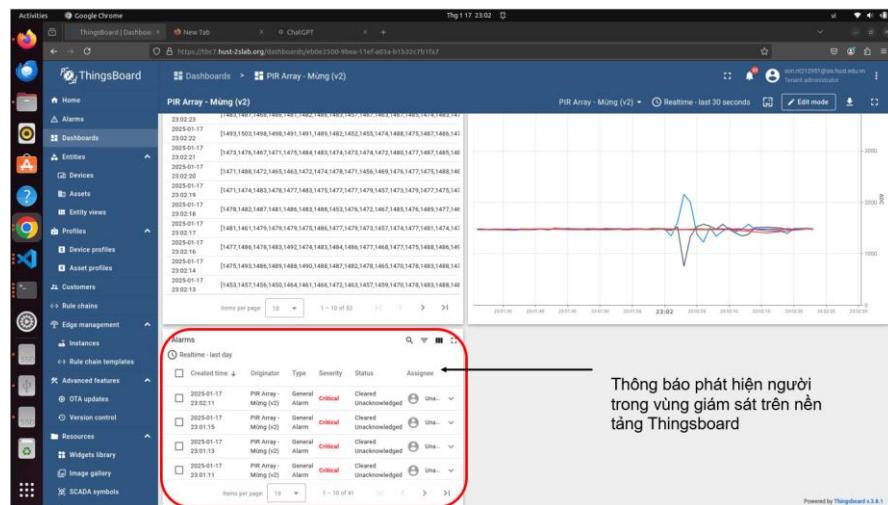


Hình 4.5 Đồ thị trực quan trên Thingsboard khi phát hiện chuyển động trong vùng giám sát



Hình 4.6 Xét ngưỡng phát hiện người cho cảm biến

Trên hình là dạng tín hiệu của một cảm biến khi phát hiện chuyển động trong vùng giám sát. Ở đây em xét 2 ngưỡng là có phát hiện người và chưa có phát hiện. Nếu tín hiệu xoay quanh giá trị (1300-1600), em xét ngưỡng đó là chưa có chuyển động trong vùng giám sát. Còn khi giá trị nằm ngoài mức này thì là có phát hiện chuyển động trong vùng giám sát. Từ đó mà trên nền tảng IoT Thingsboard có thể đưa ra thông báo là phát hiện người, phục vụ cho việc ra lệnh bật/tắt các thiết bị ngoại vi trong nhà nếu cần.



Hình 4.7 Thông báo phát hiện người trên Thingsboard

Khi tín hiệu WiFi mạnh, tức là khi node cảm biến được đặt gần nguồn mạng, dữ liệu được truyền lên Thingsboard một cách đều đặn và ổn định. Tuy nhiên, khi nguồn mạng được đưa ra xa, mặc dù ESP32 vẫn duy trì kết nối và tiếp tục truyền dữ liệu, nhưng một số gói tin bị mất do sự suy giảm tín hiệu. Mặc dù vậy, hệ thống vẫn có khả năng tự động kết nối lại với WiFi trong trường hợp mất kết nối, đảm bảo việc truyền tải dữ liệu không bị gián đoạn, duy trì hoạt động liên tục của hệ thống.

Timeseries table	
🕒 Realtime - last minute	
Timestamp	top ↓
2025-01-12 11:49:23	[1713,1710,1708,1706,1701,1696,1691,1685,1677,1674,1664,16]
2025-01-12 11:49:14	[1652,1645,1635,1638,1623,1611,1597,1585,1568,1573,1569,15]
2025-01-12 11:49:21	[1568,1558,1571,1573,1579,1578,1579,1579,1571,1556,1558,15]
2025-01-12 11:49:25	[1565,1562,1562,1562,1566,1567,1568,1577,1584,1585,1585,15]
2025-01-12 11:49:16	[1534,1523,1520,1519,1522,1511,1506,1517,1511,1514,1518,15]
2025-01-12 11:49:22	[1529,1520,1516,1520,1527,1534,1521,1522,1529,1527,1527,15]
2025-01-12 11:49:27	[1527,1534,1531,1531,1535,1535,1535,1535,1535,1535,1535,15]

Hình 4.8.Các gói tin khi kết nối tốt

Timeseries table	
🕒 Realtime - last minute	
Timestamp	top ↓
603,1605,1605,1600,1600,1600,1601,1602,1605,1602,1605,1605,1605,1600,16	
637,1635,1635,1644,1644,1645,1642,1644,1641,1639,1645,1647,1646,1645,16	
742,1744,1746,1741,1743,1735,1738,1739,1735,1736,1731,1724,1727,1712,1712,17	

Hình 4.9.Mất gói tin khi kết nối kém

-Quy trình truyền tin:

+Khởi tạo kết nối TCP: Client (ESP32) thực hiện một TCP 3-way handshake với server (ThingsBoard). Trong quá trình này, client và server trao đổi các gói tin SYN, SYN-ACK, và ACK để thiết lập kết nối TCP.

+Truyền dữ liệu: Sau khi kết nối TCP được thiết lập, ESP32 bắt đầu truyền gói tin HTTP (chứa thông tin như label, timestamp và dữ liệu các giá trị ADC) tới ThingsBoard

+Xử lý trên server ThingsBoard: Server nhận gói tin, thực hiện xử lý và phản hồi lại với client (ESP32) bằng một thông báo ACK hoặc kết quả.

4.3 Thủ nghiệm thuật toán với dữ liệu thu được cho việc triển khai tinyML trong tương lai

Quá trình huấn luyện và đánh giá được chạy trên nền tảng phần cứng là laptop TUF A15 với CPU AMD RYZEN 7 6000 SERIES; GPU RTX 3060 6GB VRAM; 16GB RAM DDR5 và nền tảng phần mềm là jupyter notebook, python kernel 3.10, CUDA V11.8.89.

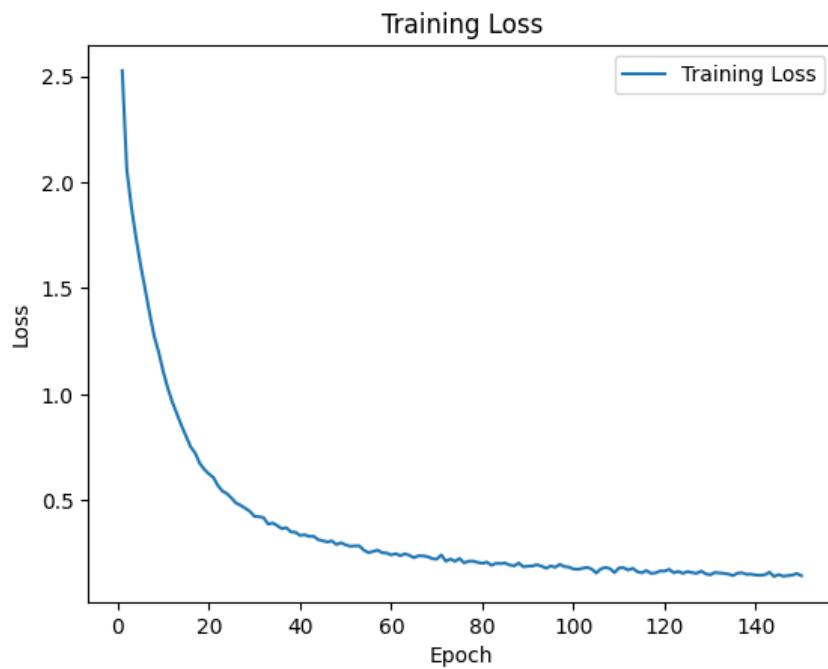
-Mô hình định vị theo nhãn

Mô hình được huấn luyện với thông số như sau:

- Num epochs = 200
- Hàm loss là CrossEntropyLoss()
- Thuật toán optimization là Adam với weight decay = 0.0001 và Learning rate = 0.0001

Kết quả:

Hình dưới là đồ thị training loss sau 200 vòng lặp training với 80% bộ train



Hình 4.10 Training Loss

Sau khi cho mô hình đã train xử lý 20% bộ train còn lại và tính số nhãn dự đoán đúng trên tổng số nhãn dự đoán ta thu được kết quả 86.41%. Kết quả dự đoán theo từng nhãn được thể hiện ở bảng dưới

Label	Precision	Recall	F1 Score
11	0.86	0.89	0.88
12	0.84	0.89	0.86
13	0.83	0.88	0.85
14	0.93	0.86	0.89
21	0.82	0.84	0.83
22	0.91	0.84	0.87
23	0.85	0.91	0.88
24	0.88	0.86	0.87
31	0.87	0.85	0.86
32	0.85	0.84	0.84
33	0.85	0.89	0.87
34	0.83	0.88	0.85
41	0.94	0.85	0.89
42	0.87	0.92	0.9
43	0.8	0.88	0.84
44	0.88	0.89	0.89
51	0.89	0.89	0.89
52	0.9	0.78	0.84
53	0.84	0.88	0.86
54	0.85	0.96	0.9
61	0.84	0.93	0.88
62	0.79	0.75	0.77

63	0.94	0.77	0.85
64	0.88	0.83	0.85
71	0.89	0.93	0.91
72	0.91	0.85	0.88
73	0.84	0.93	0.88
74	0.9	0.86	0.88
81	0.87	0.85	0.86
82	0.9	0.85	0.88
83	0.86	0.87	0.87
84	0.89	0.89	0.89
91	0.85	0.88	0.86
92	0.88	0.82	0.85
93	0.88	0.85	0.87
94	0.75	0.81	0.78

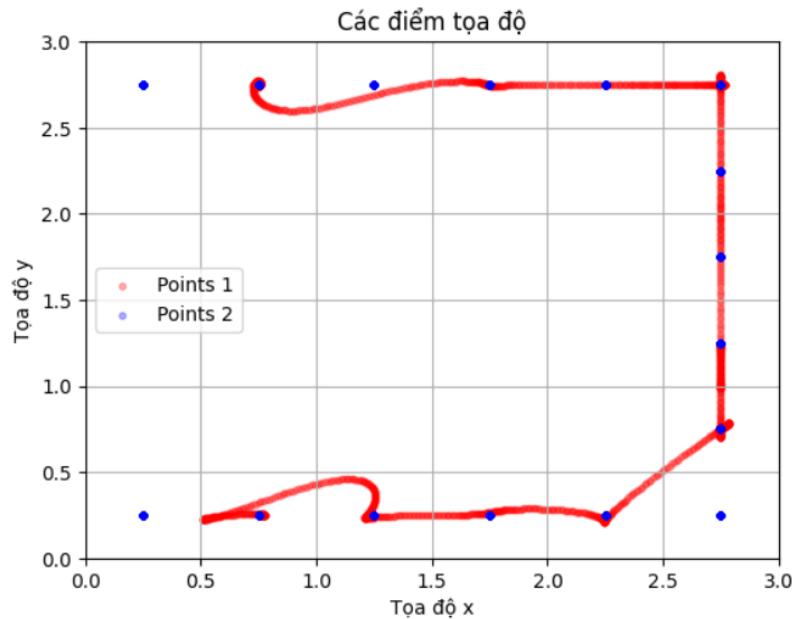
Bảng 2 Kết quả dự đoán theo từng nhãn

Dưới đây là kết quả thu được khi cho mô hình xử lý dữ liệu trong các kịch bản trong bộ test

Kịch bản 1:

Lọc thông thấp đầu ra tần số cắt 1Hz

Kappa score: 0.4043424915787225
Khoảng cách trung bình: 0.1468700123965936

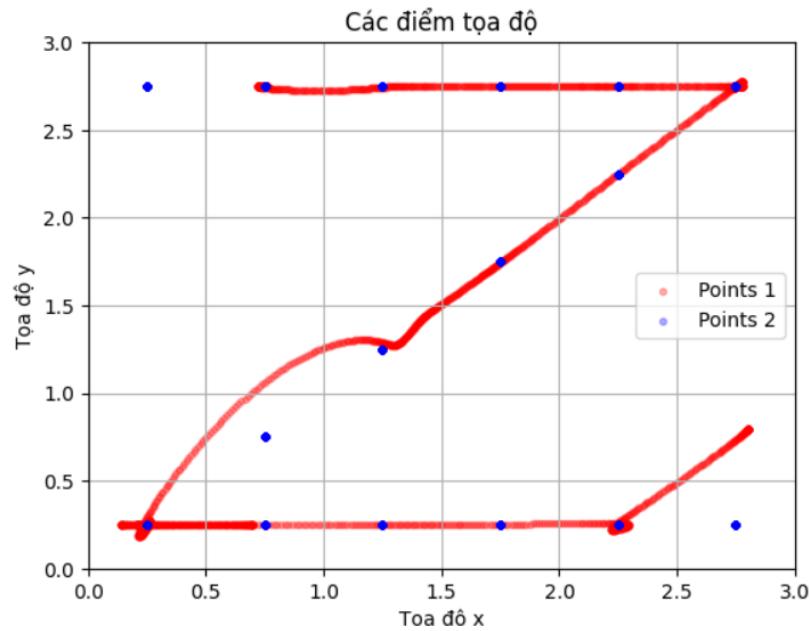


Hình 4.11 Kết quả kịch bản 1

Kịch bản 2:

Lọc thông thấp đầu ra tần số cắt 0.8Hz

Kappa score: 0.6879696606091075
Khoảng cách trung bình: 0.09328939314982201

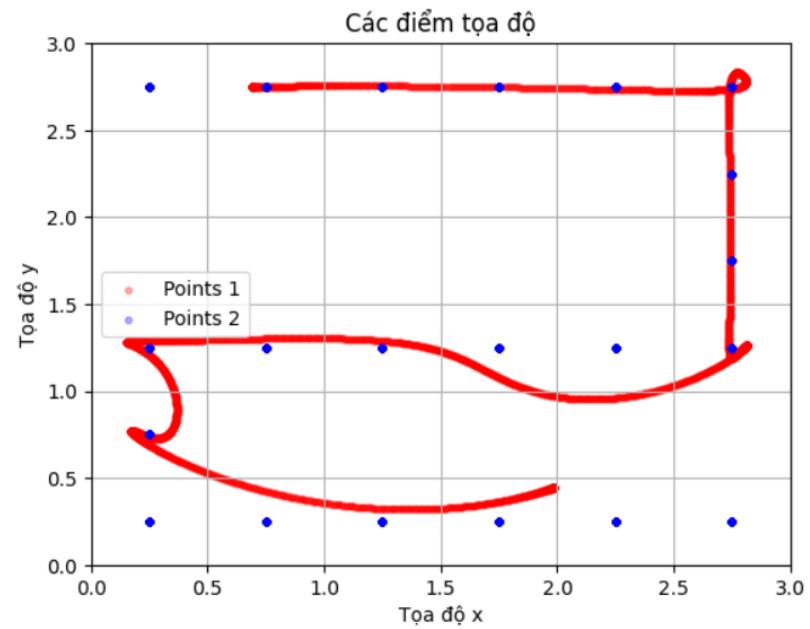


Hình 4.12 Kết quả kịch bản 2

Kịch bản 3:

Lọc thông thấp đầu ra tần số cắt 0.2Hz

Kappa score: 0.569125465545359
Khoảng cách trung bình: 0.15105025178897422

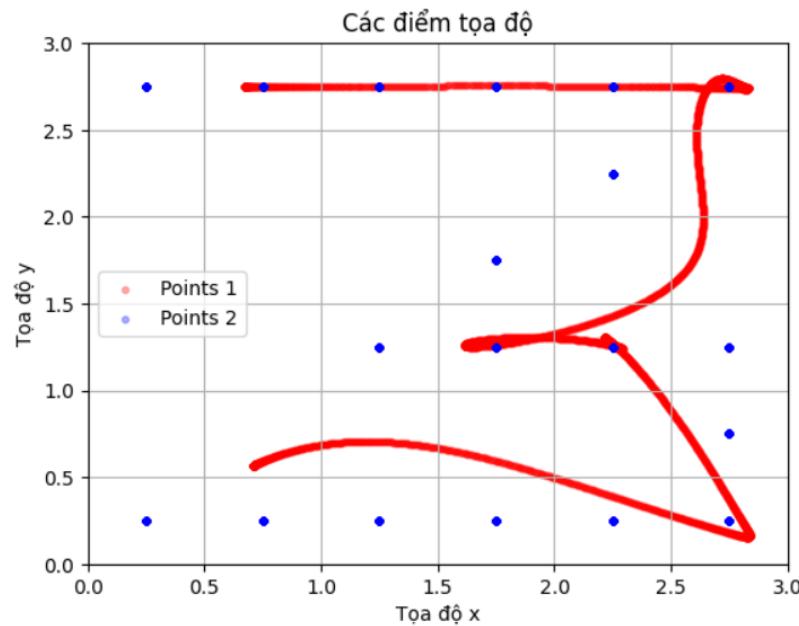


Hình 4.13 Kết quả kịch bản 3

Kịch bản 4:

Lọc thông thấp đầu ra tần số cắt 0.2Hz

Kappa score: 0.46742678062084364
Khoảng cách trung bình: 0.183142408115931



Hình 4.14 Kết quả kịch bản 4

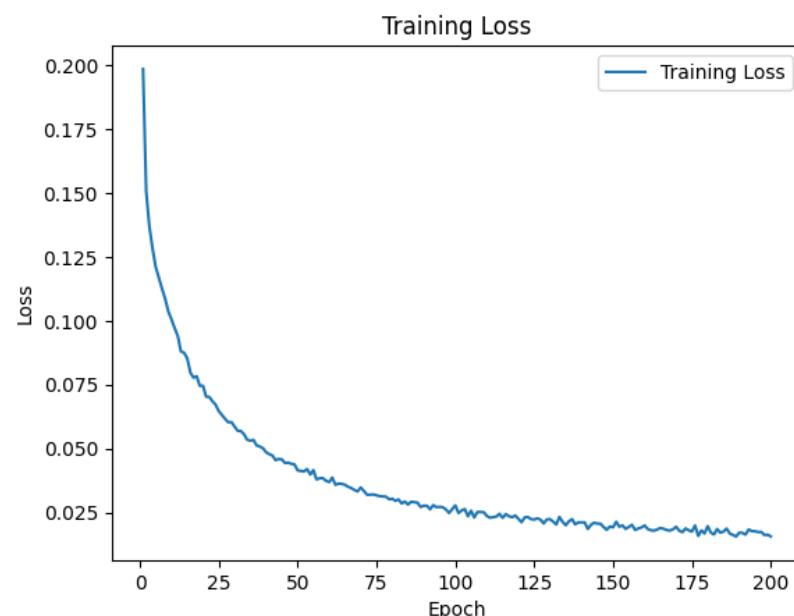
-Mô hình định vị theo tọa độ

Mô hình được huấn luyện với thông số như sau:

- Num epochs = 200
- Hàm loss là MSELoss()
- Thuật toán optimization là Adam với weight decay = 0.0001 và Learning rate = 0.0001

Kết quả

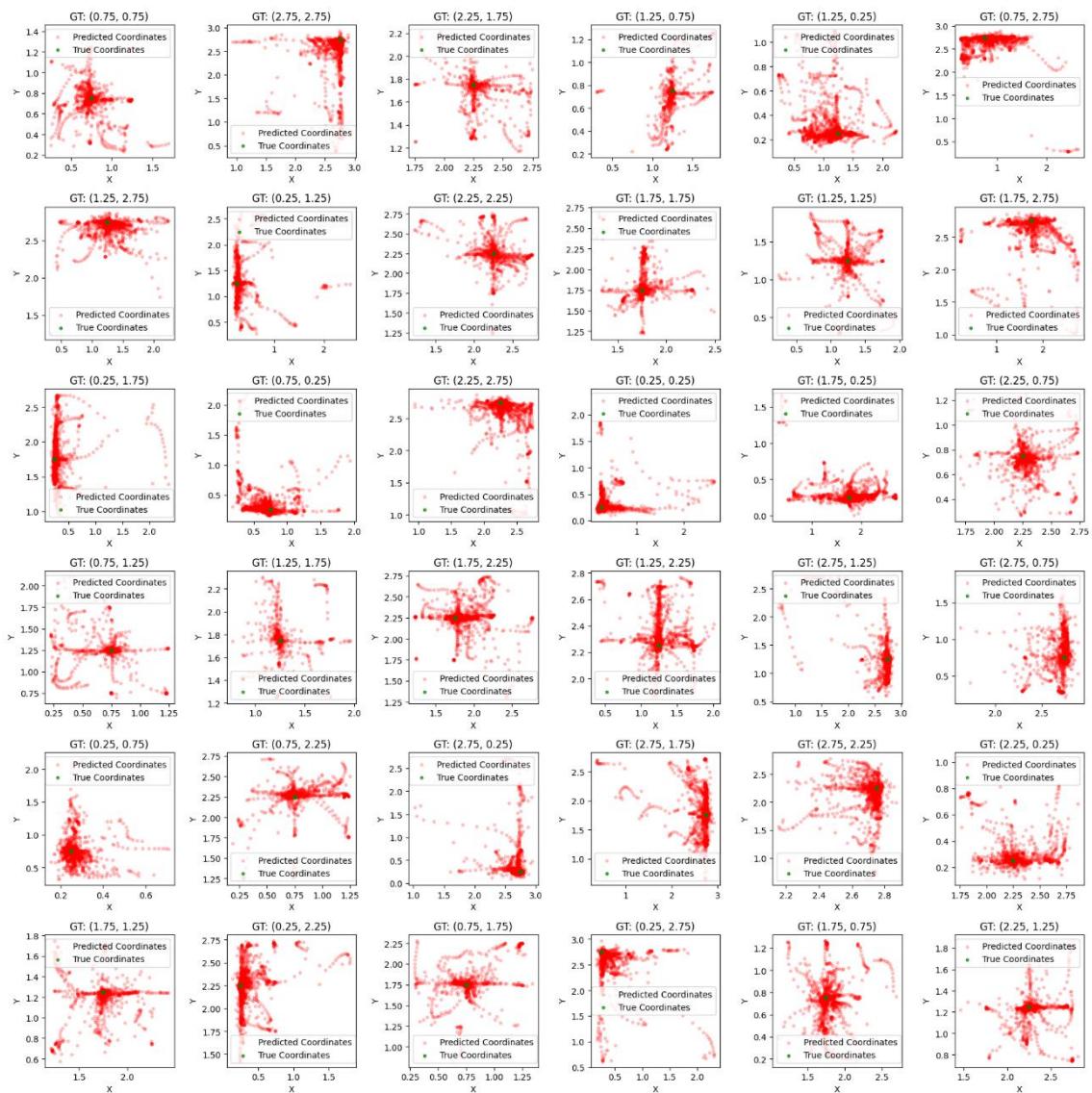
Hình dưới là đồ thị training loss sau 200 vòng lặp với 80% bộ train:



Hình 4.15 Training Loss

Sau khi cho mô hình đã train xử lý 20% bộ train còn lại và tính số vị trí dự đoán có sai số trong bán kính 0.35m trên tổng số vị trí dự đoán ta thu được kết quả 82.23%. Sai số khoảng cách trung bình là 0.1817m

Phân佈 vị trí dự đoán so với nhãn đúng của bộ dữ liệu có thể quan sát ở hình dưới



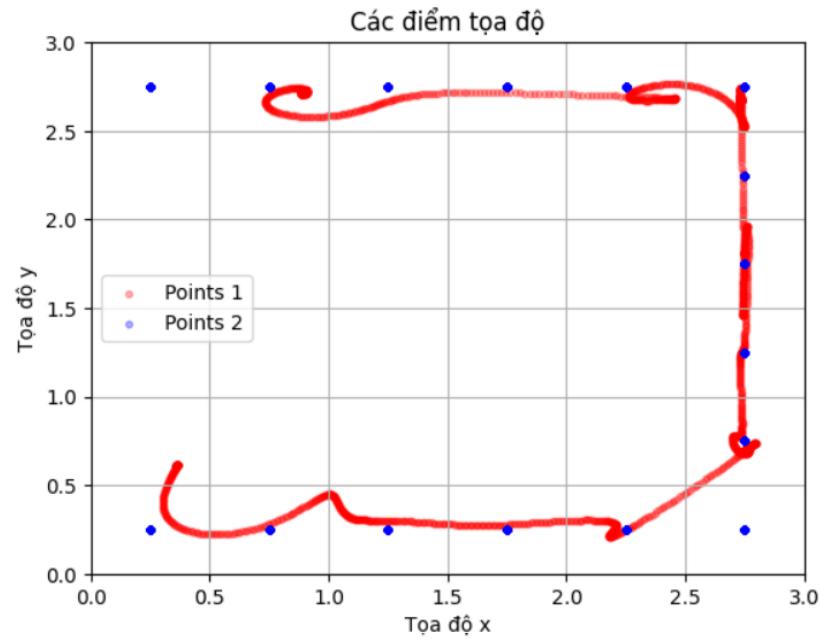
Hình 4.16 Phân佈 dự đoán vị trí

Dưới đây là kết quả thu được khi cho mô hình xử lý dữ liệu trong các kích thước trong bộ test

Kích thước 1:

Lọc thông thấp đầu ra tần số cắt 1Hz

Khoảng cách trung bình: 0.28720593442528136

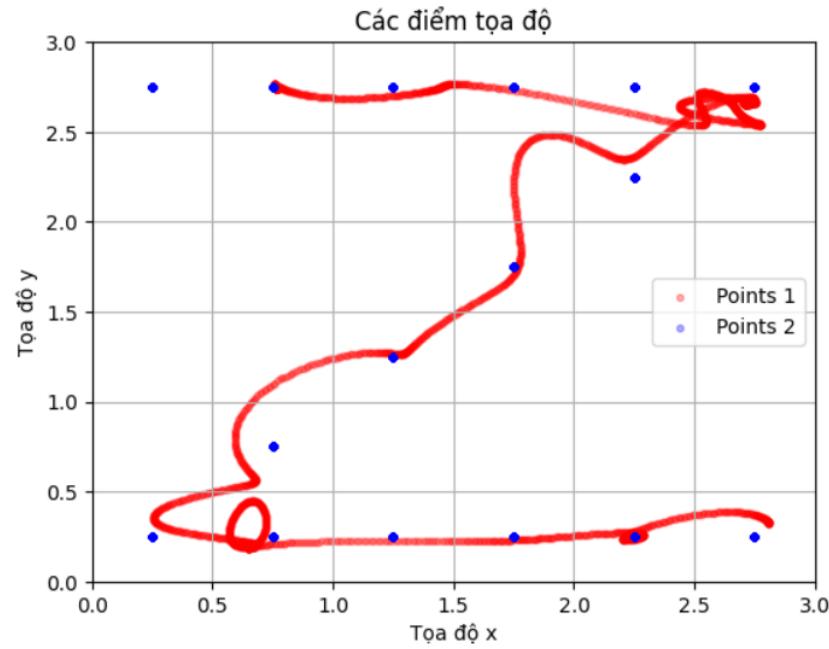


Hình 4.17 Kết quả kịch bản 1

Kịch bản 2:

Lọc thông thấp đầu ra tần số cắt 0.8Hz

Khoảng cách trung bình: 0.2622858475161426

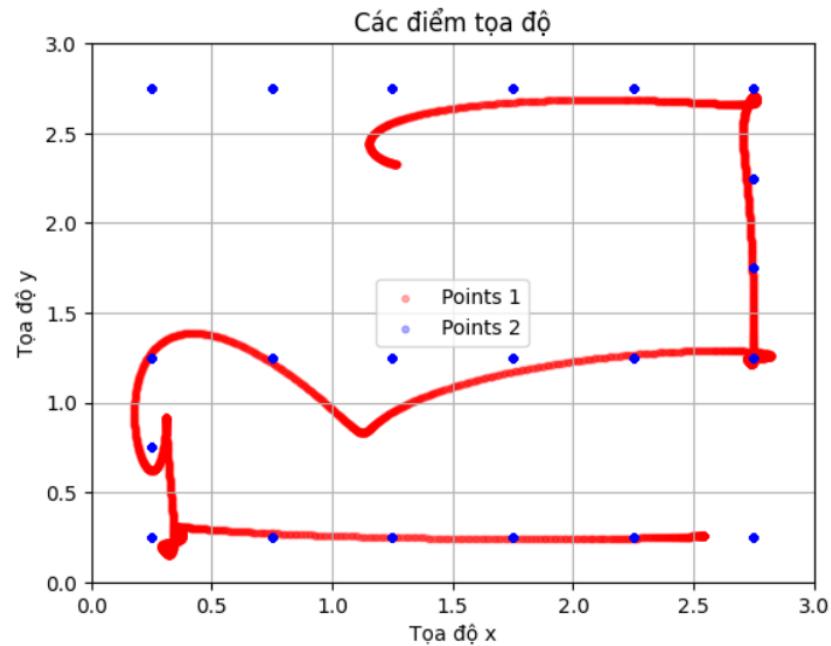


Hình 4.18 Kết quả kịch bản 2

Kịch bản 3:

Lọc thông thấp đầu ra tần số cắt 0.3Hz

Khoảng cách trung bình: 0.25553329777204786

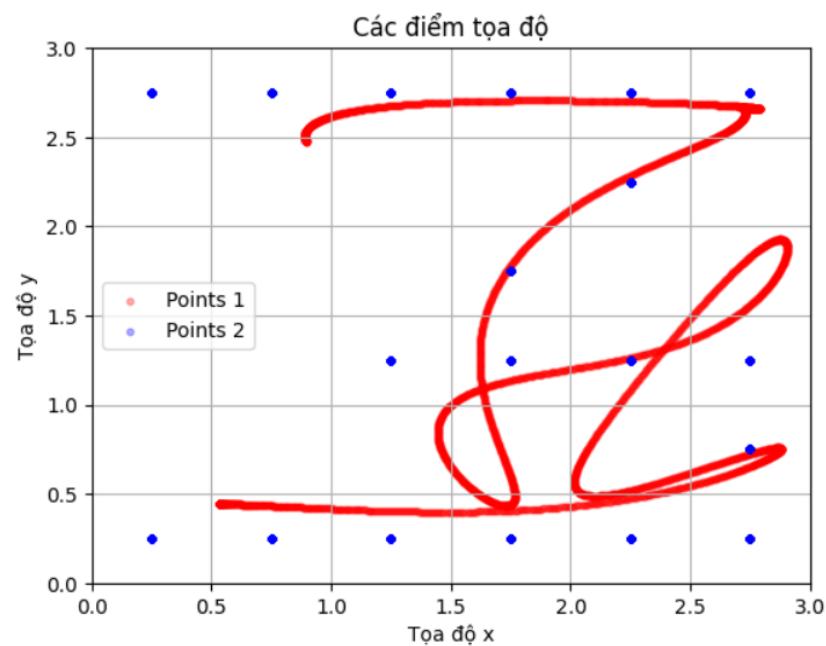


Hình 4.19 Kết quả kịch bản 3

Kịch bản 4:

Lọc thông thấp đầu ra tần số cắt 0.2Hz

Khoảng cách trung bình: 0.3784866364299632



Hình 4.20 Kết quả kịch bản 4

4.4 Đánh giá hệ thống tổng thể

Sau quá trình thử nghiệm phần cứng và phần mềm IoT, hệ thống cảm biến phát hiện chuyển động sử dụng ESP32 và cảm biến PIR đã cho thấy hiệu quả và tiềm

năng ứng dụng trong các bài toán giám sát. Hệ thống hoạt động ổn định, với cảm biến PIR có khả năng phát hiện chuyển động chính xác trong phạm vi bán kính 3m khi được lắp đặt ở độ cao 2.8m. Điện áp đầu ra của cảm biến cho thấy sự thay đổi rõ rệt giữa trạng thái có và không phát hiện chuyển động, giúp hệ thống dễ dàng nhận diện các tín hiệu bất thường. Đồng thời, việc duy trì độ nhạy ở mức thấp giúp loại bỏ hiện tượng bão hòa tín hiệu, đảm bảo độ chính xác cao.

Kết nối WiFi thông qua ESP32 hỗ trợ việc truyền dữ liệu liên tục lên nền tảng Thingsboard. Các giá trị từ cảm biến được trực quan hóa bằng biểu đồ và bảng Timeseries, giúp người dùng dễ dàng theo dõi và phân tích sự thay đổi tín hiệu theo thời gian. Hệ thống cũng được tích hợp chức năng tự động kết nối lại WiFi khi mất tín hiệu, đảm bảo tính liên tục trong truyền tải dữ liệu. Tuy nhiên, khi tín hiệu WiFi yếu, một số gói tin vẫn bị mất, ảnh hưởng đến khả năng phân tích dữ liệu.

Hệ thống hiện vẫn tồn tại một số hạn chế, bao gồm phạm vi giám sát còn hạn chế ở mức 3m và phụ thuộc nhiều vào chất lượng tín hiệu WiFi. Ngoài ra, tín hiệu đầu ra từ cảm biến PIR chỉ dùng lại ở việc phát hiện chuyển động cơ bản.

Để tính toán năng lượng tiêu thụ của node cảm biến, chúng ta sẽ xét các thành phần chính của hệ thống, bao gồm 5 cảm biến PIR và ESP32, cùng với các hoạt động của chúng.

Mỗi cảm biến PIR tiêu thụ dòng điện 0.29mA khi hoạt động ở điện áp 3.3V.



Hình 4.21 Dòng điện của mỗi cảm biến

Công suất tiêu thụ của một cảm biến có thể tính toán bằng công thức:

$$P_{PIR} = I_{PIR} * V_{cc} = 0.29 * 10^{-3} * 3.3 = 0.957 \text{ mW}$$

Với 5 cảm biến PIR, tổng công suất tiêu thụ của chúng là:

$$P_{5PIR} = 5 * 0.957 = 4.785 \text{ mW}$$

Dòng tiêu thụ khi ESP32 truyền dữ liệu qua WiFi : $I_{wifi} = 250 \text{ mA}$

Điện áp $V_{ESP32} = 3.3 \text{ V}$

Công suất tiêu thụ trong trạng thái truyền:

$$P_{wifi} = I_{wifi} * V_{ESP32} = 0.25 * 3.3 = 0.825W$$

Mỗi gói tin có thời gian truyền xấp xỉ 0.1s, nên tổng thời gian truyền 5 gói tin là 0.5s. Lúc này, thời gian ở trạng thái chờ sẽ là 0.5s. Công suất của ESP32 ở trạng thái chờ là $P_{idle} = 0.066W$. Do đó, công suất trung bình trong 1s sẽ là:

$$P_{ESP32_avg} = 0.5 * P_{wifi} + 0.5 * P_{idle} * 0.5 = 0.825 * 0.5 + 0.066 * 0.5 = 0.4455(W)$$

Tổng công suất tiêu thụ của toàn bộ node cảm biến:

$$P_{total} = P_{ESP32_avg} + P_{5PIR} = 0.4455 + 0.004785 = 0.450285(W)$$

Dánh giá mức tiêu thụ năng lượng của toàn bộ node cảm biến cho thấy hệ thống hoạt động với mức tiêu thụ năng lượng khá hợp lý và hiệu quả, đáp ứng tốt các yêu cầu của các ứng dụng IoT cần tiết kiệm năng lượng. Cảm biến PIR, với mức tiêu thụ năng lượng chỉ khoảng 0.957 mW mỗi cảm biến, đóng góp một phần nhỏ vào tổng mức tiêu thụ năng lượng của hệ thống. Khi sử dụng 5 cảm biến PIR, tổng công suất tiêu thụ của các cảm biến này chỉ đạt 4.785 mW, cho thấy rằng các cảm biến PIR rất phù hợp cho các ứng dụng yêu cầu tiết kiệm năng lượng, nhờ vào mức tiêu thụ năng lượng thấp và khả năng hoạt động liên tục mà không làm ảnh hưởng nhiều đến nguồn năng lượng.

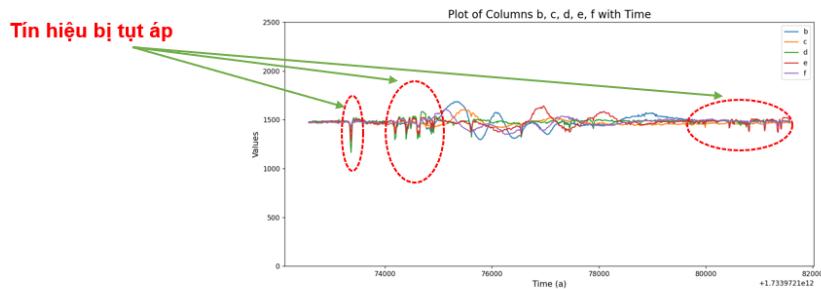
Hệ thống có khả năng mở rộng và tích hợp với nhiều thiết bị ngoại vi trong hệ sinh thái Smarthome thông qua nền tảng Thingsboard. Việc truyền dữ liệu qua WiFi cho phép hệ thống kết nối linh hoạt với các thiết bị ngoại vi như đèn, quạt, điều hòa, cửa cuốn, máy bơm, và các thiết bị tự động khác. Hệ thống có thể tự động điều khiển các thiết bị dựa trên dữ liệu cảm biến, như tắt đèn khi không có chuyển động, điều chỉnh nhiệt độ khi có người trong phòng, hoặc mở cửa khi phát hiện sự di chuyển. Với khả năng tự kết nối lại WiFi khi mất tín hiệu, hệ thống duy trì ổn định và dễ dàng tích hợp thêm nhiều thiết bị mới để phục vụ các nhu cầu Smarthome trong tương lai.

4.5 Khó khăn

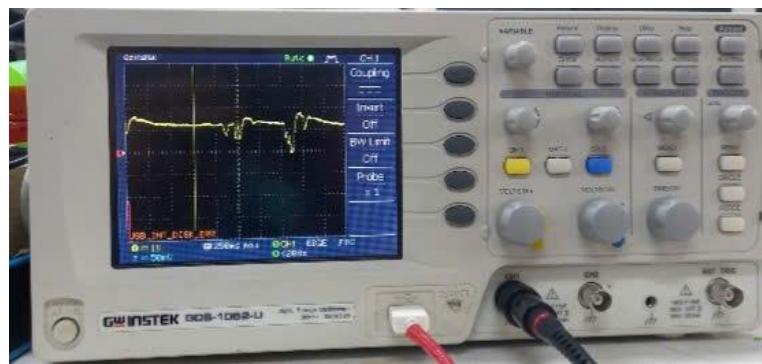
Trong quá trình thực nghiệm, một hiện tượng đáng chú ý xảy ra là tín hiệu thu được từ cảm biến PIR thông qua ADC thường xuyên bị tụt xuống hoặc không ổn định. Sau khi tiến hành phân tích chi tiết và loại trừ các nguyên nhân khác, nguyên nhân chính được xác định là do cảm biến bị ảnh hưởng bởi nhiều sóng điện từ (EMI - Electromagnetic Interference). Đây là một vấn đề phổ biến trong các hệ thống điện tử, đặc biệt là những hệ thống nhạy cảm với tín hiệu như mạch cảm biến và ADC.

Nhiều sóng điện từ (EMI) phát sinh từ nhiều nguồn khác nhau trong môi trường, chẳng hạn như các thiết bị phát sóng Wi-Fi, Bluetooth, động cơ điện, bộ nguồn xung (switching power supply), và thậm chí là các vi mạch điện khác hoạt động gần đó. Các thiết bị này phát ra sóng điện từ với tần số cao, và các sóng này có thể tác động trực tiếp đến tín hiệu truyền từ cảm biến đến ADC của vi điều khiển.

Trong trường hợp cụ thể của cảm biến PIR, tín hiệu đầu ra của cảm biến thường có biên độ nhỏ, dễ bị ảnh hưởng bởi các nhiễu bên ngoài. Điều này khiến tín hiệu trở nên méo hoặc bị tụt, làm giảm độ chính xác trong quá trình thu thập dữ liệu.



Hình 4.22 Tín hiệu bị nhiễu do sóng điện từ



Hình 4.23 Tín hiệu output của cảm biến đo được khi bị nhiễu

Một trong những nguyên nhân chính khiến nhiễu EMI ảnh hưởng mạnh đến hệ thống là do thiếu các biện pháp chống nhiễu hiệu quả. Các dây dẫn tín hiệu giữa cảm biến và mạch PCB thường hoạt động như những anten nhỏ, dễ dàng thu nhận sóng điện từ từ môi trường xung quanh. Nếu không được thiết kế với các giải pháp chống nhiễu phù hợp hoặc không được bố trí gần mặt đất (GND) của mạch, các dây dẫn này sẽ dễ dàng truyền sóng điện từ vào tín hiệu ADC. Hậu quả là giá trị ADC đo được có thể bị sai lệch, không phản ánh chính xác tín hiệu thực tế từ cảm biến.

4.6 Phân tích chi phí và tiềm năng triển khai

Chi phí sản xuất của hệ thống này rất thấp nhờ vào việc sử dụng các cảm biến và linh kiện có giá thành rẻ và dễ dàng tìm thấy trên thị trường. Cảm biến PIR, một trong những thành phần chính của hệ thống, có giá rất hợp lý, khoảng 10.000 VND, và được bán rộng rãi trên các sàn thương mại điện tử như Shopee, Lazada,...hoặc các cửa hàng điện tử, giúp giảm chi phí đầu tư ban đầu. ESP32, với khả năng tích hợp WiFi và Bluetooth, cũng là một lựa chọn tiết kiệm, có sẵn trên thị trường với mức giá phải chăng. Các linh kiện phụ như điện trở, tụ điện, và IC LM358 đều dễ dàng tìm thấy tại các cửa hàng điện tử với giá thành rẻ, góp phần vào việc tối ưu hóa chi phí sản xuất của hệ thống.

Về tiềm năng ứng dụng thực tế, hệ thống này có thể được ứng dụng rộng rãi trong nhiều lĩnh vực. Trong môi trường Smarthome, hệ thống có thể kết nối và điều khiển các thiết bị như đèn, quạt, điều hòa, cửa cuốn, giúp tối ưu hóa việc sử dụng năng lượng và tạo ra một không gian sống tiện nghi. Bên cạnh đó, hệ thống còn có thể ứng dụng trong các lĩnh vực an ninh và giám sát, giúp phát hiện chuyển động và cảnh báo về các xâm nhập trái phép. Với khả năng mở rộng và tích hợp dễ dàng, hệ thống còn có tiềm năng trong các ứng dụng chăm sóc sức khỏe, quản lý năng lượng trong các tòa nhà thông minh, cũng như giám sát các khu vực công cộng. Với chi phí sản xuất thấp và tiềm năng ứng dụng cao, hệ thống này có thể đem lại giá trị thực tế lớn và dễ dàng triển khai trong nhiều lĩnh vực khác nhau.

CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1 Kết luận

Đề tài "Thiết kế và triển khai hệ thống cảm biến PIR ứng dụng trong Smarthome" đã hoàn thành các mục tiêu đề ra. Hệ thống được thiết kế với phần cứng ổn định, sử dụng ESP32 để thu thập và truyền dữ liệu qua Wi-Fi đến nền tảng Thingsboard, giúp giám sát và lưu trữ dữ liệu thời gian thực. Hệ thống hoạt động hiệu quả trong các ứng dụng Smarthome với độ chính xác cao trong việc phát hiện chuyển động.

Ưu điểm của hệ thống là chi phí thấp, dễ lắp đặt và mở rộng, nhưng vẫn còn hạn chế trong việc phát hiện chuyển động phức tạp và phạm vi hoạt động nhỏ. Trong tương lai, hệ thống có thể được tích hợp thuật toán TinyML để phân loại chuyển động, mở rộng quy mô thành mạng cảm biến lớn hơn và ứng dụng vào các lĩnh vực như an ninh hoặc quản lý năng lượng.

5.2 Hướng phát triển trong tương lai

Mặc dù dự án cảm biến PIR Array đã đạt được nhiều kết quả tích cực, hệ thống vẫn còn rất nhiều tiềm năng để tiếp tục nâng cấp và mở rộng nhằm đáp ứng các nhu cầu ngày càng cao trong giám sát và phân tích dữ liệu. Các hướng phát triển dưới đây sẽ giúp dự án trở nên toàn diện, hiệu quả và tiên tiến hơn.

Nâng cao độ chính xác và khả năng nhận diện

Một hướng quan trọng là tích hợp trí tuệ nhân tạo (AI) và học máy (Machine Learning) để phân tích dữ liệu từ cảm biến. Điều này không chỉ giúp tăng độ chính xác mà còn mở rộng khả năng nhận diện đối tượng. Chẳng hạn, AI có thể giúp phân loại và phát hiện các loại đối tượng khác nhau trong khu vực giám sát như người, vật nuôi hoặc các vật thể di chuyển khác. Bên cạnh đó, việc cải thiện thuật toán lọc nhiễu sẽ giúp giảm thiểu các tín hiệu sai lệch do ánh sáng, nhiệt độ hoặc các nguồn nhiễu môi trường không mong muốn. Đặc biệt, khả năng phát hiện nhiều người đồng thời trong cùng một khu vực sẽ là một bước tiến lớn trong việc nâng cấp tính năng của hệ thống.

Mở rộng phạm vi giám sát

Hệ thống có thể mở rộng bằng cách tăng số lượng cảm biến PIR, tạo ra mạng lưới giám sát rộng lớn hơn, đặc biệt hữu ích trong các không gian lớn như nhà máy, trung tâm thương mại hoặc các khu vực công cộng. Ngoài ra, thiết kế cảm biến đa tầng, hoạt động trên các độ cao khác nhau, sẽ giúp giám sát hiệu quả trong không gian ba chiều, chẳng hạn như các tòa nhà cao tầng. Bên cạnh cảm biến PIR, tích hợp các loại cảm biến khác như siêu âm, ánh sáng hoặc hình ảnh sẽ giúp hệ thống thu thập dữ liệu phong phú và đa dạng hơn.

Tối ưu hóa hiệu suất hệ thống

Một hướng phát triển quan trọng khác là tăng cường hiệu suất vi điều khiển bằng cách sử dụng các vi điều khiển mạnh mẽ hơn với bộ nhớ lớn hơn và khả năng xử lý nhanh hơn. Đồng thời, các chiến lược tiết kiệm năng lượng, chẳng hạn như sử

dụng chế độ ngủ cho cảm biến khi không có chuyển động, sẽ giúp tăng thời gian hoạt động của hệ thống, đặc biệt trong các ứng dụng sử dụng pin. Ngoài ra, việc sử dụng kiến trúc server phân tán hoặc điện toán đám mây sẽ giúp xử lý dữ liệu từ nhiều cảm biến một cách hiệu quả hơn, tránh tình trạng quá tải.

Triển khai thuật toán trên MCU với TinyML

Việc ứng dụng TinyML sẽ mở ra cơ hội triển khai các mô hình học máy trực tiếp trên vi điều khiển ESP32. Điều này không chỉ giảm sự phụ thuộc vào server mà còn tăng tính bảo mật và giảm độ trễ trong xử lý dữ liệu. TinyML cũng cho phép hệ thống tự học và thích nghi với môi trường hoạt động, mang lại khả năng tự chủ cao hơn. Đây là một giải pháp quan trọng để nâng cấp toàn bộ hệ thống, từ xử lý tín hiệu cho đến phản hồi thông minh.

Phát triển giao diện và báo cáo thông minh

Giao diện người dùng (UI) có thể được nâng cấp để thân thiện hơn, cho phép hiển thị dữ liệu qua các biểu đồ, bảng, hoặc bản đồ trực quan, giúp người dùng dễ dàng theo dõi và quản lý hệ thống. Ngoài ra, việc tích hợp các tính năng cảnh báo như gửi thông báo qua email, SMS hoặc ứng dụng di động khi phát hiện hoạt động bất thường sẽ giúp người dùng phản ứng nhanh hơn với các tình huống khẩn cấp.

Tăng cường tính bảo mật và ổn định

Để đảm bảo an toàn dữ liệu, các thuật toán mã hóa hiện đại cần được áp dụng, đặc biệt trong việc truyền dữ liệu giữa cảm biến và server. Hơn nữa, hệ thống cần tích hợp các cơ chế tự phát hiện và khắc phục lỗi nhằm đảm bảo hoạt động ổn định, ngay cả khi gặp sự cố về phần cứng hoặc phần mềm.

Tích hợp IoT vào hệ sinh thái lớn hơn

Hệ thống cảm biến PIR Array có thể được phát triển để tương thích với các nền tảng IoT lớn như Google Cloud IoT, AWS IoT hoặc Microsoft Azure, giúp mở rộng tính năng và quy mô ứng dụng. Đồng thời, việc hỗ trợ điều khiển từ xa qua các ứng dụng di động hoặc web sẽ giúp tăng tính linh hoạt và tiện lợi cho người dùng.

Ứng dụng trong các lĩnh vực khác

Hệ thống có thể được mở rộng để ứng dụng trong nhiều lĩnh vực khác nhau. Trong quản lý năng lượng, hệ thống có thể tự động điều chỉnh ánh sáng hoặc thiết bị điện dựa trên sự hiện diện của con người, giúp tiết kiệm năng lượng. Trong chăm sóc sức khỏe, hệ thống có thể theo dõi chuyển động bất thường của bệnh nhân, hỗ trợ phát hiện các sự cố như té ngã. Ngoài ra, trong nông nghiệp, hệ thống có thể được sử dụng để giám sát hoạt động của vật nuôi hoặc phát hiện động vật hoang dã xâm nhập vào khu vực canh tác.

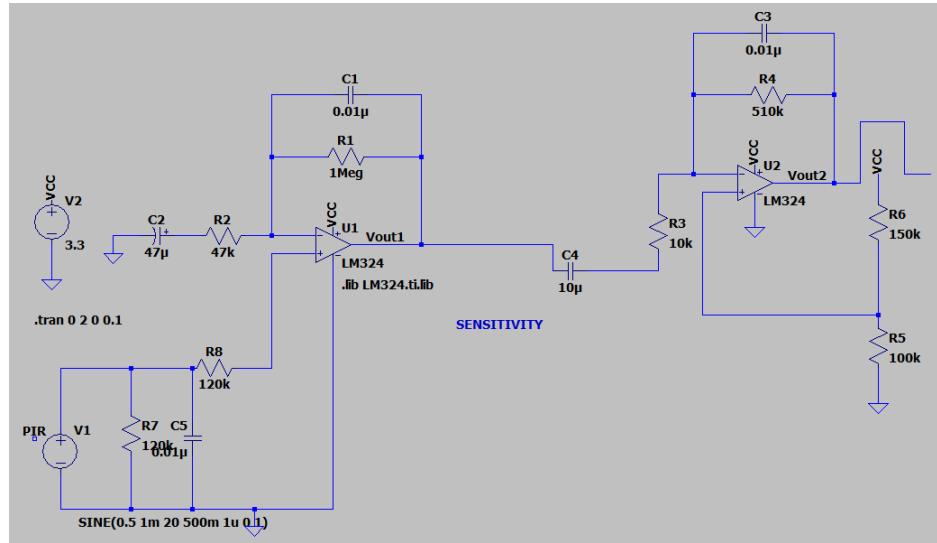
Những định hướng trên không chỉ giúp dự án trở nên toàn diện hơn mà còn mở ra cơ hội ứng dụng trong nhiều lĩnh vực khác nhau, mang lại giá trị thực tiễn cao trong đời sống và sản xuất. Việc tiếp tục nghiên cứu và phát triển các ý tưởng này sẽ đưa hệ thống cảm biến PIR Array trở thành một giải pháp tiên tiến, đáng tin cậy và bền vững trong tương lai.

TÀI LIỆU THAM KHẢO

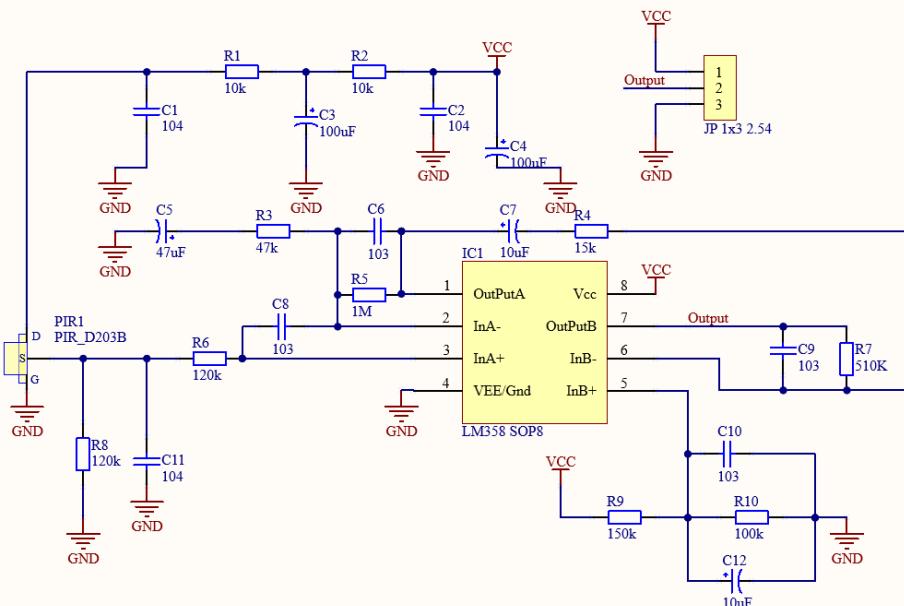
- [1] D. Semiconductor, "DB203B Datasheet," 2019.
- [2] I. t. lai, "Giao thức dữ liệu IoT," [Online]. Available: <http://iottuonglai.com/giao-thuc-du-lieu-iot.html>.
- [3] ThingsBoard, "ThingsBoard Documentation," [Online]. Available: <https://thingsboard.io/docs/>.
- [4] M. GPT, "TinyML: Ứng dụng, giới hạn và sử dụng trong thiết bị IoT và Edge Devices".
- [5] T. Instruments, "LM358 Low Power Dual Operational Amplifier Datasheet," 2004.
- [6] E. Systems, "ESP32 Technical Reference Manual," 2019.
- [7] K. Ngamakeura,S. Yongchareon,J. Yu,S. Islam, "Passive infrared sensor dataset and deep learning models for device-free indoor localization and tracking," 2021.
- [8] Nguyen Quoc Cuong,Le Minh Thuy,Hoang Ngoc Chau,Nguyen Thi Nhat Linh,Doan Tuan Kiet, "Intelligent_Passive_Infrared_Sensor_based_on_Learn," 2024.

PHỤ LỤC

A1. Sơ đồ mô phỏng



A2. Sơ đồ nguyên lý



https://github.com/Mungnv212894/datn_1